

MOUNTING METHOD, FILM ACCESS METHOD FOR DIRECTORY AND ACCESS AUTHORITY DECISION METHOD

Publication number: JP11282741 (A)

Publication date: 1999-10-15

Inventor(s): BIN KIYOUKA; MASUOKA YOSHIMASA; INOHARA SHIGEKAZU; NODA FUMIO

Applicant(s): HITACHI LTD

Classification:

- international: G06F12/00; G06F12/00; (IPC1-7): G06F12/00

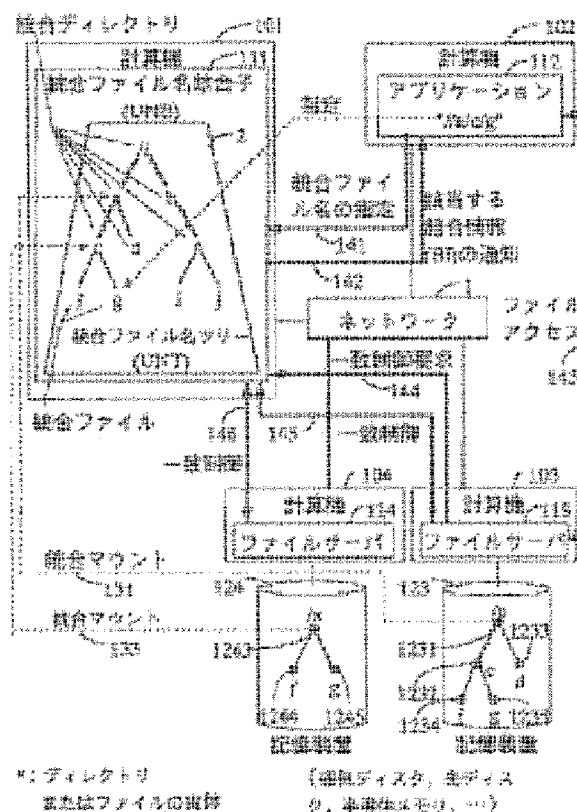
- European:

Application number: JP19980080719 19980327

Priority number(s): JP19980080719 19980327

Abstract of JP 11282741 (A)

PROBLEM TO BE SOLVED: To provide a flexible mounting method for a directory wherein a user can specify a mountable file or the like. **SOLUTION:** When a directory/x of a local file name tree is mounted on any of directory (c) of a common file name tree 2, information which a user specifies, such as mounting permission condition for indicating kinds of mountable file of the same name or the like, read-out operation information for controlling access operations, write operation information or an access authority table, is set in accordance with a mounting designation party (c). Access right is also set corresponding to a mounting source/x. Thus, selection of a file where mounting is allowed and execution of an access demand to a file (f) or (g) are controlled.



Data supplied from the **esp@cenet** database — Worldwide

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-282741

(43) 公開日 平成11年(1999)10月15日

(51) Int.Cl.⁶
G 0 6 F 12/00

識別記号
5 4 5

F I
G 0 6 F 12/00

5 4 5 B

審査請求 未請求 請求項の数12 ○L (全 34 頁)

(21) 出願番号 特願平10-80719

(22) 出願日 平成10年(1998)3月27日

(71) 出願人 000003108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 関 京華

東京都国分寺市東恋ヶ窪一丁目280番地

株式会社日立製作所中央研究所内

(72) 発明者 増岡 義政

東京都国分寺市東恋ヶ窪一丁目280番地

株式会社日立製作所中央研究所内

(72) 発明者 猪原 茂和

東京都国分寺市東恋ヶ窪一丁目280番地

株式会社日立製作所中央研究所内

(74) 代理人 弁理士 高橋 明夫 (外1名)

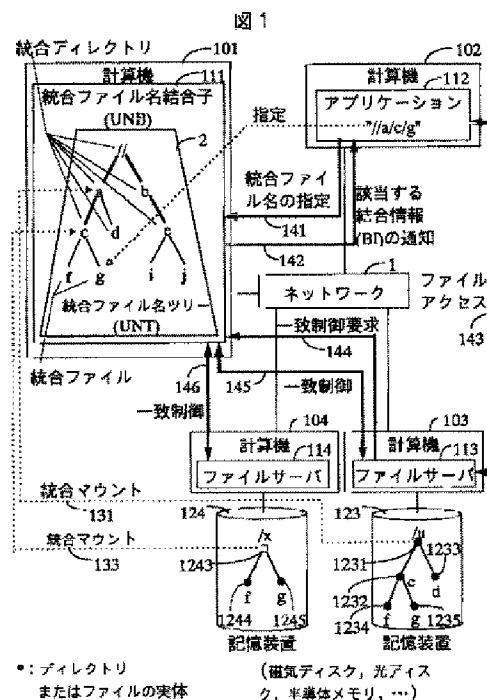
最終頁に続く

(54) 【発明の名称】 ディレクトリのマウント方法、ファイルアクセス方法およびアクセス権限判別方法

(57) 【要約】

【課題】 マウント可能なファイルなどをユーザが指定できる柔軟な、ディレクトリのマウント方法を提供する。

【解決手段】 ローカルファイル名ツリーのディレクトリ/xを共通ファイル名ツリー2のいずれかのディレクトリcにマウントするときに、マウント可能な同名ファイルの種別などを示すマウント許可条件、アクセス動作を規制する読み出し操作情報、書き込み操作情報およびアクセス権限表などユーザが指定する情報をマウント先cに対応して設定する。アクセス権限はマウント元/xにも対応して設定される。これらにより、マウントを許可されるファイルの選択と、マウントされたファイルfまたはgに対するアクセス要求の実行が制御される。



【特許請求の範囲】

【請求項1】ネットワークにより接続された複数の計算機と、それぞれ上記複数の計算機のいずれかに接続された複数の記憶装置とを有し、それぞれ上記複数の記憶装置の一つに記憶された複数のファイルが属する複数の共通ディレクトリが登録された、上記複数の計算機に共通のファイル名ツリーと、各記憶装置に記憶された複数のファイルが属する複数のローカルディレクトリが登録された、各記憶装置に対応するローカルファイル名ツリーとを使用して各記憶装置に記憶されたファイルをアクセスする分散計算機システムにおいて、
上記共通のファイル名ツリーに含まれる複数のディレクトリの各々に対応して、そのディレクトリに属することができるファイルに関連する、マウントを要求したユーザが指定したマウント条件を予め記憶し、
上記複数の記憶装置のいずれかに対応するローカルファイル名ツリーのいずれかのマウント元のローカルディレクトリを、上記共通のファイル名ツリーのいずれかのマウント先の共通ディレクトリにマウントする前に、上記マウント元のローカルディレクトリに属する複数のファイルの各々が、そのマウント先の共通のディレクトリに対応して記憶されたマウント条件を満たすか否かを判別し、
上記複数のファイルの内、上記マウント先の共通ディレクトリに対応して記憶された上記マウント条件を満たすファイルが上記マウント先の共通のディレクトリに新たに属するように、上記マウント元のローカルディレクトリを上記マウント先の共通のディレクトリにマウントするディレクトリのマウント方法。

【請求項2】上記共通のファイル名ツリーのいずれかの共通ディレクトリに対応して記憶された上記マウント条件は、その共通ディレクトリにすでに属するファイルとその共通ディレクトリに新たに属しうるファイルとの関係に関する条件を含む請求項1記載のディレクトリのマウント方法。

【請求項3】上記マウント条件は、その共通ディレクトリにすでに属するファイルの名称と同じ名称を有する同名ファイルに関する条件を含む請求項2記載のディレクトリのマウント方法。

【請求項4】上記マウント条件は、上記同名ファイルの種類に関する条件を含む請求項3記載のディレクトリのマウント方法。

【請求項5】上記同名ファイルの種類は、複製ファイル、同等ファイル、同類ファイルの内の少なくとも一つを含む請求項4記載のディレクトリのマウント方法。

【請求項6】上記マウント元のローカルディレクトリを上記マウント先の共通ディレクトリにマウントする前に、上記マウント元のローカルディレクトリに対応して、マウント元が複製ディレクトリか否かを指定し、
上記マウント元のローカルディレクトリが複製ディレク

トリとして指定されなかったときには、上記判別ステップと上記マウントステップとを実行し、
上記マウント元のローカルディレクトリが複製ディレクトリとして指定されたときには、マウント先の共通のディレクトリにすでに属しているファイルの写しを上記マウント元のローカルディレクトリに転送する請求項1記載のディレクトリのマウント方法。

【請求項7】ネットワークにより接続された複数の計算機と、それぞれ上記複数の計算機のいずれかに接続された複数の記憶装置とを有し、それぞれ上記複数の記憶装置の一つに記憶された複数のファイルが属する複数の共通ディレクトリが登録された、上記複数の計算機に共通のファイル名ツリーと、各記憶装置に記憶された複数のファイルが属する複数のローカルディレクトリが登録された、各記憶装置に対応するローカルファイル名ツリーとを使用して各記憶装置に記憶されたファイルをアクセスする分散計算機システムにおいて、
いずれかの共通ディレクトリに、いずれかの記憶装置に対するローカルファイル名ツリーのいずれかのローカルディレクトリをマウントするときに、マウント元のローカルディレクトリに属するファイルが同名ファイルであるときには、そのファイルに関する情報として、そのファイルの同名ファイルの種類を上記共通ディレクトリに対応して記憶し、

上記マウントのときに、その共通ディレクトリに属する複数の同名ファイルがある場合にそれらの同名ファイルのいずれか一つを選択して読み出すための条件を指定する、マウントを要求したユーザにより指定された読み出し操作情報をその共通のディレクトリに対応して記憶し、

その共通ディレクトリに属するいずれかのファイルに対する読み出し要求が発生したときに、その共通ディレクトリにマウントされたマウント元のローカルディレクトリが複数あれば、それらのマウント元のローカルディレクトリに属するファイルの中に、アクセス要求が指定したファイル名と同じ名称を有する複数の同名ファイルがあるか否かを、それぞれのファイルに対して記憶されたファイルの種類により判別し、

複数の同名のファイルがあれば、その共通ディレクトリに対応して記憶された上記読み出し操作情報に従って、上記複数の同名ファイルの一つを選択して読み出すファイルアクセス方法。

【請求項8】上記同名ファイルの種類は、複製ファイル、同等ファイル、同類ファイルの内の少なくとも一つを含む請求項7記載のファイルアクセス方法。

【請求項9】ネットワークにより接続された複数の計算機と、それぞれ上記複数の計算機のいずれかに接続された複数の記憶装置とを有し、それぞれ上記複数の記憶装置の一つに記憶された複数のファイルが属する複数の共通ディレクトリが登録された、上記複数の計算機に共通

のファイル名ツリーと、各記憶装置に記憶された複数のファイルが属する複数のローカルディレクトリが登録された、各記憶装置に対応するローカルファイル名ツリーとを使用して各記憶装置に記憶されたファイルをアクセスする分散計算機システムにおいて、

いずれかの共通ディレクトリに、いずれかの記憶装置に対するローカルファイル名ツリーのいずれかのローカルディレクトリをマウントするときに、マウント元のローカルディレクトリに属するファイルが同名ファイルであるときには、そのファイルに関する情報として、そのファイルの同名ファイルの種類を上記共通ディレクトリに対応して記憶し、

上記マウントのときに、その共通ディレクトリに属する複数の同名ファイルがある場合にそれらの同名ファイルに対して実行すべき書き込み操作を指定する、上記マウントを要求したユーザにより指定された書き込み操作情報をその共通ディレクトリに対応して記憶し、

その共通ディレクトリに登録されたいずれかのファイルに対する書き込み要求が発生したときに、その共通ディレクトリにマウントされたマウント元のローカルディレクトリが複数あれば、それらのマウント元のローカルディレクトリに属するファイルの中に、アクセス要求が指定したファイル名と同じ名称を有する複数の同名ファイルがあるかを、それぞれのファイルに対して記憶されたファイルの種類により判別し、

複数の同名のファイルがあれば、その共通ディレクトリに対応して記憶された上記書き込み操作情報に従って、上記複数の同名ファイルに対して上記書き込み要求に関連する書き込み操作を実行するファイルアクセス方法。

【請求項10】上記同名ファイルの種類は、複製ファイル、同等ファイル、同類ファイルの内の少なくとも一つを含む請求項9記載のファイルアクセス方法。

【請求項11】ネットワークにより接続された複数の計算機と、それぞれ上記複数の計算機のいずれかに接続された複数の記憶装置とを有し、それぞれ上記複数の記憶装置の一つに記憶された複数のファイルが属する複数の共通ディレクトリが属する、上記複数の計算機に共通のファイル名ツリーと、各記憶装置に記憶された複数のファイルが属する複数のローカルディレクトリが属する、各記憶装置に対応するローカルファイル名ツリーとを使用して各記憶装置に記憶されたファイルをアクセスする分散計算機システムにおいて、

上記共通のファイル名ツリーに含まれた複数の共通ディレクトリの各々に対応して、その共通ディレクトリに登録されたいずれかのファイルのアクセスを許可するか否かを決定するための、ユーザが指定した共通ディレクトリアクセス権限情報を予め記憶し、

いずれかの共通ディレクトリに属するファイルに対するアクセス要求が発生したときに、そのアクセス要求にそのファイルへのアクセスを許すか否かを、その共通ディ

レクトリに対して記憶された共通ディレクトリアクセス権限情報に基づいて判別するアクセス権限判別方法。

【請求項12】いずれかの共通ディレクトリに、いずれかの記憶装置に対するローカルファイル名ツリーのいずれかのローカルディレクトリをマウントするときに、その共通ディレクトリに対応して、そのローカルディレクトリに属するファイルのアクセスを許可するか否かを決定するための、ユーザが指定したローカルディレクトリアクセス権限情報を記憶するステップをさらに有し、上記判別ステップでは、上記アクセス要求にアクセスを許すか否かを、上記アクセス要求が要求するファイルが属する共通ディレクトリに関して記憶された共通ディレクトリアクセス権限情報と、その共通ディレクトリにマウントされたローカルディレクトリに関して記憶されたローカルディレクトリアクセス権限情報とに基づいて判別する請求項11記載のアクセス権限判別方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は分散ファイルシステムのディレクトリを合成する方法に関する。

【0002】

【従来の技術】計算機のファイル・システムにおける名前づけは、ファイル・システムの基本機能のひとつである。名前づけの役割は、計算機上のさまざまなデータやプログラム（以後、ファイルと総称する）を一意に識別することと、ユーザ間のファイルの共有を助けることである。ファイルにつけられた名前を「ファイル名」と呼ぶ。その全体を「ファイル名空間」と呼ぶ。

【0003】多くのファイル・システムのファイル名空間は、階層的な木構造をなす。この木構造は、ディレクトリとファイルからなる。ディレクトリは木構造の分岐点である。一方、ファイルは木構造の末端であり、ディレクトリの下におかれる。このようなファイル名空間をファイル名ツリーとも呼ぶ。

【0004】ネットワークによって、ひとつの計算機から別の計算機上のファイルをアクセスすることが可能になった。1980年頃から、ネットワーク上の複数の計算機を跨る分散ファイル・システムの技術が登場し、発展してきた。このなかの代表的な技術には、共通ファイル名空間の技術、位置透過性と独立性の技術、キャッシング技術、複製技術、アクセス・コントロール技術、認証技術があり、これらは現在も発展を続けている。以下、共通ファイル名空間の技術、位置透過性と位置独立性の技術を説明する。

【0005】R. Sandberg他著“The Sun Network File System: Design, Implementation and Experience”, A. L. AnandaとB. Srinivasan編“Distributed Computing Systems: Concepts and Structures”, IEEE Computer Society Press, pp. 300-314, 1991に記載の方法（以後「公知例1」と呼ぶ）によれば、計算機Bのファイル・

システムにより管理されるネームツリー内のいずれかのディレクトリ、例えばXに、他の計算機Aのファイルシステムにより管理されるネームツリー内の他のディレクトリ、例えばUを組み込むという操作が知られている。この操作では、ディレクトリXの名称をディレクトリUに割り当てる。この操作を行うと、計算機B上のアプリケーションプログラムが計算機AのディレクトリUにアクセスする場合、アクセス要求がディレクトリXを指定すればよく、計算機AのディレクトリUをユーザが指定する必要はない(位置透過性)。

【0006】このように、あるネームツリー上のディレクトリXの名称を他のネームツリー上のディレクトリUに割り当てる操作は、ディレクトリXにディレクトリUをマウントすると呼ばれている。この操作では、ディレクトリXは、マウント先のディレクトリと呼ばれ、ディレクトリUは、マウント元のディレクトリと呼ばれる。

【0007】J. H. Howard他著“Scale and Performance in a Distributed File System”, ACM Transactions on Computer Systems, vol. 6, no. 1, pp. 51-81, February 1988に記載の方法(以後「公知例2」と呼ぶ)では、異なるファイル・システムが提案されている。ファイル・サーバ計算機とクライアント計算機を区別しており、各クライアント計算機は、ファイル名空間をローカルなファイル名空間(もしくはローカルなファイル名ツリー)と共通ファイル名空間(もしくは共通ファイル名ツリー)とで構成する。本明細書を通じて「ローカル」という語は、ひとつの計算機がネットワークを経由せずにアクセスできる資源を指す。共通ファイル名空間もしくは共通ファイル名ツリーはファイル・サーバ計算機群が提供し、ローカルなファイル名空間もしくはローカルなファイル名ツリーは各クライアント計算機がそれぞれ別のものを持つ。共通ファイル名ツリーを構成するディレクトリおよびローカルなファイル名ツリーを構成するディレクトリを本明細書では、共通ディレクトリおよびローカルディレクトリと呼ぶことがある。この従来技術では、異なるクライアント計算機が管理するローカルディレクトリを共通のファイルネームツリー上の共通ディレクトリにマウントされる。各クライアント計算機は、このような共通ネームツリーを使用してそれぞれのサーバ計算機が管理する共通ファイルを、それぞれのファイルを管理するサーバ計算機がどれであるかを意識しないでアクセスすることが出来る。また、クライアント計算機側に影響せず、共通ネームツリーに登録されたディレクトリやファイルをファイル・サーバ計算機群の中に移動することが可能である。このように、ディレクトリやファイルが特定の計算機に縛られないことを、以後「位置独立性」と呼ぶ。

【0008】以上の公知例に記載のマウント方法では、各計算機に散在したファイル(以後「分散ファイル」と呼ぶ)を論理的にまとめた時に次の問題が生じる。た

えば、同じ種類の情報を格納したファイルが別々の計算機のローカルなファイル名ツリーに属する二つのディレクトリに入っている場合、一方のローカルファイルネームツリー上のローカルなディレクトリを他方のローカルファイルネームツリー上の他のローカルディレクトリにまとめて整理したい場合、上記一方のディレクトリを上記他方のディレクトリにマウントすると、上記一方のディレクトリに属する複数のファイルをアクセスすることができるようになるが、上記一方のディレクトリに属する複数のファイルは隠されてしまい、アクセスすることができなくなる。

【0009】このような問題を有しないマウント方法として、ユニオン・マウント法が提案されている。この技術によれば、上記の問題が無いマウントの結果、マウント元のディレクトリはマウント先のディレクトリに合成されると呼んでいる。ユニオン・マウント法には二種類がある。M. K. McKusick他著“The Design and Implementation of the 4.4 BSD Operating System”, Addison-Wesley, 1996の6.7 Stackable Filesystemsに記載の方法(以後「公知例3」と呼ぶ)とR. Pike他著“The Use of Name Space in Plan 9”, Plan 9: Programmer's Manual Volume 2, Second Edition, pp. 29-34, 1995に記載の方法(以後「公知例4」と呼ぶ)である。

【0010】

【発明が解決しようとする課題】これらの方法によるディレクトリのマウントは異なるところがあるが、マウントされるディレクトリに上下関係が付けられる。すなわち、あるディレクトリ(マウント先のディレクトリ)に他のディレクトリ(マウント元のディレクトリ)をマウントするとき、マウント先のディレクトリはマウント元のディレクトリの下側のディレクトリとして扱われる。ディレクトリのマウント結果はその上下関係に依存する。即ち、ユーザがあるファイル名のファイルに対するアクセスを試みたとき、そのファイル名のファイルが属するディレクトリと他のディレクトリとの上下関係により、そのファイルへのアクセスあるいはそのファイルの更新が制限されるなどファイルが属するディレクトリの上下関係に依存してそのファイルへのアクセスに制限が生じる。以下、図2を用いて具体的に説明する。

【0011】図2では、記憶装置20にあるディレクトリ“x”21、ファイル“a”22、ファイル“b”23、サブディレクトリ“z”24、ファイル“d”25が格納されており、記憶装置30にあるディレクトリ“y”31、ファイル“a”32、ファイル“c”33、サブディレクトリ“z”34、ファイル“e”35が格納されている。公知例3と公知例4のユニオン・マウント法をそれぞれ使って、記憶装置20にあるディレクトリ“x”21を記憶装置30にあるディレクトリ“y”31の上にマウントするとする。

【0012】公知例3のユニオン・マウントのマウント

結果はファイル名ツリー40に示される。ファイル名ツリー40にあるディレクトリ“y”41はマウント・ポイントであり、ディレクトリ“x”21をそこにマウントする前は記憶装置30にあるディレクトリ“y”31を参照する。ディレクトリ“x”21をそこにマウントすれば、次のマウント仕様に従ってマウントが行われる。ここで、マウント先のディレクトリ“y”41は、マウント元のディレクトリ“x”21より下に位置するディレクトリとして扱われる。

【0013】(a) 上側のディレクトリ“x”21に含まれるファイル“a”22と下側のディレクトリ“y”41に含まれるファイル“a”32は同名のファイルである。下側のディレクトリ“y”31に属する、記憶装置30内のファイル“a”32は参照されないし、変更もされない。このように、下側ディレクトリのファイルは、上側ディレクトリの同名のファイルにより隠される。なお、ファイル“a”42は、記憶装置20内のファイル“a”22を参照する。また、ファイル“a”42を変更すれば(48)参照の、記憶装置20内のファイル“a”22が変更される(48')。

【0014】(b) 記憶装置30内のファイル“c”33と同名のファイルは、上側ディレクトリ“x”21に含まれていない。このようにマウント対象となったディレクトリ群の中に一番上ではないディレクトリに入っているファイルはread-onlyに制限される。すなわち、ファイル“c”44は記憶装置30内のファイル“c”33を参照できるが、これを変更することができない。

【0015】(c) もしファイル“c”44を変更しようとする(49)、ファイル“c”33は、記憶装置20にある一番上側のディレクトリ“x”21にコピーアップされる(49')。すなわち、記憶装置20内のファイル“c”33'が新規に作られ、その後は、変更は記憶装置20内のファイル“c”33'に反映される(49'')。記憶装置30内のファイル“c”33にはこの変更は反映されない。

【0016】(d) 上側のディレクトリ“x”21に含まれるサブディレクトリ“z”24と下側のディレクトリ“y”41に含まれるサブディレクトリ“z”34は同名のサブディレクトリである。ファイル名ツリー40では、サブディレクトリ“z”45において、ディレクトリ“y”41におけるマウントと同じマウント仕様に従ったマウントがサブディレクトリ“z”34とサブディレクトリ“z”24とに対して再帰的に行われる。しかし、ファイル名ツリー40内のファイル“d”46は、記憶装置30内のファイル“e”35を参照するが、この参照はread-onlyである。

【0017】公知例4によるマウント結果はファイル名ツリー50に示される。ファイル名ツリー50にあるディレクトリ“y”51はマウント・ポイントであり、マ

ウントする前に記憶装置30にあるディレクトリ31を参照する。一旦そこにマウントすれば次のマウント仕様に従ってマウントが行われる。

【0018】(a) 同名ファイルは全部見えるが、そのうちの一番上のファイルにしかアクセスできない。すなわち、ファイル名ツリー50にファイル“a”52とファイル“a”52'が二つ見える。ファイル“a”52は記憶装置20内のファイル“a”22を参照する。ファイル“a”52'は参照先がない。ファイル“a”52を変更すれば(58)、参照先記憶装置20内のファイル“a”22が変更される(58')。

【0019】(b) 同名のサブディレクトリは全部見えるが、そのうちの一番上のサブディレクトリにしかアクセスできない。このことが原因となって、ディレクトリマウントは同名サブディレクトリに対して再帰的に行われない。すなわち、ファイル名ツリー50にサブディレクトリ“z”55とサブディレクトリ“z”55'が二つに見える。サブディレクトリ“z”55は記憶装置20内のサブディレクトリ“z”24を参照する。サブディレクトリ“z”55'は参照先がない。ファイル“d”56は、記憶装置20内のファイル“d”25を参照する。記憶装置30内のファイル“e”35はファイル名ツリー50から見えない。

【0020】なお、従来のマウント方法では、あるマウント先のディレクトリにマウント元のディレクトリをマウントするときに、マウント可否は、マウントを要求するユーザがマウント先のディレクトリとマウント元のディレクトリに対してアクセス権限を有するか否かのチェックによつてのみ判断される。

【0021】公知例3のマウント方法はもともとソフトウェアのバージョン管理に向けるディレクトリマウントであるため、仕様上、ディレクトリの上下関係が定められている。これはソフトウェアのバージョン管理に適しているが、分散ファイルをマウントする時には、上に示されたように、最も上のディレクトリ内の同名ファイルあるいはサブディレクトリ内のファイルしかアクセスあるいは更新できないのでは、ファイル管理の上で不便である。

【0022】したがって、本発明の目的は、マウントすべき複数のディレクトリの上限関係に依存しないでアクセスできるファイルあるいは更新できるファイルが定まるディレクトリマウント方法を提供することである。

【0023】本発明の他の目的は、アクセスできるファイルあるいは更新できるファイルをより柔軟に定めることができるディレクトリマウント方法を提供することである。

【0024】本発明のさらに他の目的は、アクセスできるファイルあるいは更新できるファイルをユーザが自由に定めることができるディレクトリマウント方法を提供することである。

【0025】

【課題を解決するための手段】上記課題を解決するために、本発明によるディレクトリのマウント方法では、ディレクトリのマウントに際し、マウント先のディレクトリに対応してマウント許可条件をユーザに指定可能にする。このマウント許可条件は、マウント可能なファイルに関する条件を指定する。たとえば、操作の対象となるファイル（特に同名によって重なったファイル）の内容と属性により、マウント可能なファイルを指定する。マウント許可条件を変えることによって異なる方法でディレクトリのマウントを行える。

【0026】すなわち、本発明によるマウント方法では、マウントを実行する前に、分散計算機システム内の複数の計算機に共通に定められたファイル名ツリーを構成する複数のディレクトリの各々に対応して、そのディレクトリに属することができるファイルに関連する、マウントを要求したユーザが指定したマウント条件を予め記憶する。本明細書では、ディレクトリに属するファイルはそのディレクトリの直下にあるファイル及び再帰的にそのディレクトリのサブディレクトリに属するファイルからなる集合にあるファイル、即ち、そのディレクトリにおけるサブツリーに属するファイルを指す。

【0027】分散計算機システム内の複数の計算機の一つにそれぞれ接続された複数の記憶装置のいずれかに対応するローカルファイル名ツリーのいずれかのマウント元のローカルディレクトリを、上記共通ファイル名ツリーのいずれかのマウント先の共通ディレクトリにマウントするときに、上記マウント元のローカルディレクトリに属する複数のファイルの各々が、そのマウント先の共通のディレクトリに対応して記憶されたマウント条件を満たすか否かを判別する。

【0028】上記複数のファイルの全てが上記マウント先の共通ディレクトリに対応して記憶された上記マウント条件を満たすときには、上記マウント元のローカルディレクトリを上記マウント先の共通のディレクトリにマウントする。

【0029】たとえば、同名ファイルとして、複製ファイル、同等ファイル、同類ファイルを扱う場合に、これらの同名ファイルの内のいずれに対してマウントを許可するかをユーザの指定に従ってディレクトリ毎に定めることができる。

【0030】さらに、本発明によるファイルアクセス方法では、ある共通ディレクトリに属する同じ名称を有する複数の同名ファイルのファイル名を指定するファイル読み出し要求が発生した場合に、それらの同名ファイルのいずれをアクセスするかを決めるための読み出し操作を予めその共通のディレクトリに対応してユーザが指定することを可能にする。これにより、それらの同名ファイルへのアクセスをユーザが指定した基準により行うことが可能になる。例えば、読み出し要求の処理時には、

同名によって重なった複数のファイルにはどれにでも読み出しのためのアクセスさせることもできる。また、書き込み要求の処理時には、基本ファイルに対して行った書き込みに対応する変更を複数の同名ファイルに対して、それぞれのファイルの種類に応じて行い、それによりそれらの同名ファイルと基本ファイルとの一致制御を自動的に行うことができる。

【0031】すなわち、本発明による読み出しのためのファイルアクセス方法では、いずれかの共通ディレクトリに、いずれかの記憶装置に対するローカルファイル名ツリーのいずれかのローカルディレクトリをマウントするときに、マウント元のローカルディレクトリに属するファイルが同名ファイルであるときには、そのファイルに関する情報として、そのファイルの同名ファイルの種類を上記共通ディレクトリに対応して記憶する。

【0032】上記マウントのときに、その共通ディレクトリに属する複数の同名ファイルがある場合にそれらの同名ファイルのいずれか一つを選択して読み出すための条件を指定する、マウントを要求したユーザにより指定された読み出し操作情報をその共通のディレクトリに対応して記憶する。

【0033】その共通ディレクトリに属するいずれかのファイルに対する読み出し要求が発生したときに、その共通ディレクトリにマウントされたマウント元のローカルディレクトリが複数あれば、それらのマウント元のローカルディレクトリに属するファイルの中に、アクセス要求が指定したファイル名と同じ名称を有する複数の同名ファイルがあるか否かを、それぞれのファイルに対して記憶されたファイルの種類により判別し、複数の同名のファイルがあれば、その共通ディレクトリに対応して記憶された上記読み出し操作情報に従って、上記複数の同名ファイルの一つを選択して読み出す。

【0034】また、本発明による書き込みのためのファイルアクセス方法では、いずれかの共通ディレクトリに、いずれかの記憶装置に対するローカルファイル名ツリーのいずれかのローカルディレクトリをマウントするときに、マウント元のローカルディレクトリに属するファイルが同名ファイルであるときには、そのファイルに関する情報として、そのファイルの同名ファイルの種類を上記共通ディレクトリに対応して記憶し、上記マウントのときに、その共通ディレクトリに属する複数の同名ファイルがある場合にそれらの同名ファイルに対して実行すべき書き込み操作を指定する、上記マウントを要求したユーザにより指定された書き込み操作情報をその共通のディレクトリに対応して記憶する。

【0035】その共通ディレクトリに属するいずれかのファイルに対する書き込み要求が発生したときに、その共通ディレクトリにマウントされたマウント元のローカルディレクトリが複数あれば、それらのマウント元のローカルディレクトリに属するファイルの中に、アクセス

要求が指定したファイル名と同じ名称を有する複数の同名ファイルがあるか否かを、それぞれのファイルに対して記憶されたファイルの種類により判別し、複数の同名のファイルがあれば、その共通ディレクトリに対応して記憶された上記書き込み操作情報に従って、上記複数の同名ファイルに対して上記書き込み要求に関連する書き込み操作を実行する。

【0036】さらに、本発明によるアクセス権限の判別方法では、マウント先のディレクトリ毎にユーザが指定したアクセス権限情報を定めることができる。さらに、マウント元のディレクトリごとに、ユーザが指定したアクセス権限情報を定めることができる。

【0037】

【発明の実施の形態】以下、本発明の実施の一形態として、本発明に係るマウント方法—統合マウント法を使用する分散ファイル・システム（「統合ファイル名システム」(Unified Name System-UNS)）を説明する。

【0038】統合ファイル名システムは各計算機により管理されるローカルファイル名ツリーに登録されたファイルをまとめて管理するファイル名ツリーを提供する。以後、統合ファイル名システムによるファイル名ツリーを「統合ファイル名ツリー」(Unified Name Tree-UNT)と呼ぶ。統合ファイル名ツリーは本構造であり、全ユーザに共通である。統合ファイル名ツリーは、分散計算機システムの各計算機により管理されるローカルファイル名ツリーに対比して、共通ファイル名ツリーと呼ぶことができる。本明細書では、共通ファイル名ツリーを構成するディレクトリおよびローカルなファイル名ツリーを構成するディレクトリをそれぞれ共通ディレクトリおよびローカルディレクトリと呼ぶことができる。したがって統合ファイル名ツリーを構成するディレクトリも共通ディレクトリと呼ぶことができる。

【0039】あるファイル名ツリーに属するディレクトリを他のファイル名ツリーに属するディレクトリにマウントすることは、マウント元のディレクトリに、マウント先のディレクトリの名称を割り当てる操作を言う。その割り当てに関する情報をマウント先のディレクトリに対応して記憶されるなどの処理が実行される。

【0040】統合ファイル名システムは統合ファイル名ツリーを階層的に分割して統合ファイル名ツリーの分散と多重化を実現する。これによってスケラビリティと信頼性の高いファイル名空間を提供することができる。

【0041】以後、統合マウント法によりディレクトリを他のディレクトリにマウントすることを「統合マウントする」ということがある。同様に、統合マウント法によるディレクトリのマウントを「ディレクトリ統合」と呼び、ディレクトリ統合による同名ファイルのマウントを「ファイル統合」と呼び、ディレクトリ統合とファイ

ル統合のためのポリシーを「統合ポリシー」と呼び、ディレクトリ統合から得たディレクトリを「統合ディレクトリ」と呼び、統合ディレクトリに入っているまたはファイル統合から得たファイルを「統合ファイル」と呼び、統合ディレクトリと統合ファイルの名前を「統合ファイル名」と呼ぶことがある。

【0042】図3を用いて、本発明のマウント法によるディレクトリのマウント手順の概略と特徴をまず説明する。図3では、記憶装置20は、分散計算機システム内の図示しないある計算機に接続された記憶装置であり、その記憶装置20に対するローカルファイル名ツリーには、ローカルディレクトリ“x”21、ファイル“a”22、ファイル“b”23、サブディレクトリ“z”24、ファイル“d”25が登録されていると仮定する。同様に、図示しない他の計算機に接続された記憶装置30に対するローカルファイル名ツリーには、ローカルディレクトリ“y”31、ファイル“a”32、ファイル“c”33、サブディレクトリ“z”34、ファイル“e”35が登録されていると仮定する。統合マウント法を使って、記憶装置20に対するディレクトリ“x”21と記憶装置30に対するディレクトリ“y”31を、統合ファイル名ツリー60上の統合ディレクトリ“u”61にマウントする。統合ディレクトリ“u”61はマウント・ポイントであり、マウントする時に作られる。

【0043】本発明によるマウント方法の利点は以下の通りである。

【0044】(a) 統合ファイル名ツリー内では、複数の同名ファイルの一つの統合ファイルとして見せることができる。しかもすべての同名ファイルを参照することができるようにすることもできる。どのファイルを参照するかは選択はユーザが予め指定した読み出し操作にしたがって自動的に決定される。例えば、統合ファイル名ツリー60の統合ファイル“a”62は記憶装置20内のファイル“a”22と記憶装置30内のファイル“a”32の両方を参照することができる。また、このような統合ファイルを変更する時に、設定された統合ポリシーによる一致制御方法に従って、当該統合ファイルが参照する同名ファイルを変更する。例えば、統合ファイル名ツリー60の統合ファイル“a”62を変更する時に、まず記憶装置20内のファイル“a”22を変更する(68')。ファイル“a”22への変更(68')が全部終わったら、記憶装置30内のファイル“a”32への一致制御(68)を行う。このため、同名ファイルを隠さないで適当なファイルを参照することもできるし、同名ファイルの一致性を自動的に行うこともできる。

【0045】(b) 統合ファイルは、統合ポリシーによって、自分の参照するファイルの元々の属性をそのまま持ち続けるか、或は属性の変更や追加をするかを選択で

きる。例えば、統合ファイル名ツリー60の統合ファイル“b”63が参照する記憶装置20内のファイル“b”23は変更できる属性があっても、統合ファイル“b”63は変更できない属性を持つことができる。このときには、統合ファイル名ツリー60からファイル“b”23を変更することができない。また、統合ファイル名ツリー60の統合ファイル“c”64が参考先の記憶装置30にファイル“c”33の変更できる属性を継承するとすれば、統合ファイル“c”64を変更することができる(69)。その変更はファイル“c”33への変更になる(69')。このように、仕様上の制限がないで、統合ポリシーを用いてファイルに柔軟なアクセスを行うことができる。

【0046】(c)ディレクトリマウントは同名サブディレクトリにも再帰的に行われる。例えば、統合ファイル名ツリー60の統合ディレクトリ“z”65は記憶装置20にあるディレクトリ“z”24と記憶装置30にあるディレクトリ“z”34がマウントされる場合、この再帰的なマウントが行われる。このため、マウント・ポイントにおける統合ディレクトリの全体ツリーをマウントすることができる。

【0047】統合マウント法の統合ポリシーはマウント・ポイント毎およびマウント対象となるディレクトリ毎に設定される。統合ディレクトリへのマウント操作やアクセス操作は設定された統合ポリシーに従って行われる。統合ポリシーによりファイルやディレクトリの属性(例えば、アクセス制限、日付、サイズ等)を変更または追加することもできる。このため、柔軟なファイル名空間を構築することができる。もちろん、統合マウント法は適当な統合ポリシーを使ってユニオン・マウントをシミュレートすることができる。

【0048】これらの手段によって上下関係のないディレクトリマウントができるようになり、かつ、従来のディレクトリマウントより高い柔軟性(即ち、各種のディレクトリマウント)を達成することができる。

【0049】統合ファイル名システムについて以下に詳細に説明する。

【0050】(A)統合ファイル名システムのハードウェア構成およびネットワーク接続

図1に、統合ファイル名システムのハードウェア構成を示す。1台以上の計算機101、102、103、104等が、ネットワーク1に接続されている。前記計算機は、それぞれプロセッサ、主記憶装置、ネットワーク・インタフェースを有し、計算機内でプログラムを実行したり、ネットワーク1を経由して他の計算機と通信することができる。また、計算機103および計算機104は、記憶装置(主記憶装置の一部、または磁気ディスク、光ディスクなどのような補助記憶装置)を有しており、計算機103および104上で実行されるプログラムは、それぞれ記憶装置123および124上のデータ

にアクセスすることができる。なお、図1では、統合ファイル名ツリーを管理する計算機として一つの計算機101がある場合を示しているが、この数は複数でもよい。後述するように、図7に、統合ファイル名ツリーを管理する計算機が複数である場合の構成を示す。以後のすべての説明は、統合ファイル名ツリーを管理する計算機が一つである場合あるいは複数である場合のいずれかに関して与えられるが、その説明は、統合ファイル名ツリーを管理する計算機が複数である場合あるいは一つである場合にもそのままあるいは適宜修正されてあてはめることができる。説明の修正が必要な場合にも、その修正は自明であるので、とくにその修正については説明を省略する。

【0051】前記ネットワーク・インタフェースのそれぞれには、一意なネットワーク・アドレス(以下「アドレス」と略す)を割り当てる。前記アプリケーションプログラム(以下、アプリケーションプログラムを単にアプリケーションと呼ぶ)112とファイル・サーバ113および114、および統合ファイル名結合子(Unified Name Binder-UNB)111の間の通信などは、このアドレスを用いて、次のように実現される。すなわち、送信する側が送り先のアドレスと、送りたいデータを格納している主記憶装置上の領域を指定して、自分のネットワーク・インタフェースに指令を送ることにより、ネットワークを経由して、前記送り先のアドレスを割り当てられたネットワーク・インタフェースによって受信され、受信する側で指定した主記憶装置上の領域に書き込まれる。

【0052】この「各ネットワーク・インタフェースに一意のアドレスを割り当てたとき、送信側が送り先のアドレスを指定することによって、送信したいデータが、指定したアドレスを割り当てられたネットワーク・インタフェースを有する装置に正しく届けられる」という機能を実現する通信方式についてのより具体的な説明は、文献W. Richard Stevens, "UNIX Network Programming", Prentice-Hall, pp.171-196に述べられている。本実施の形態においては、当該通信方式が確立しているものとし、本明細書では、当該通信方式についてのこれ以上の詳細な説明は省略する。

【0053】(B)統合ファイル名システムのソフトウェア構成

図1に、統合ファイル名システムのソフトウェア構成も示す。統合ファイル名システムは、ファイル・サーバ113、114、統合ファイル名結合子111により実現され、これによりアプリケーション112は統合ファイル名システムを利用することができる。ここで、ファイル・サーバ、統合ファイル名結合子およびアプリケーションは、それぞれ独立したプログラムであり、計算機上で実行される。また、図1では、ファイル・サーバ113、114、統合ファイル名結合子111、およびアプ

リケーション112は、それぞれ別の計算機で実行され、既に述べたようにネットワーク1を経由して互いに通信することができる。ただし、これらのプログラムの一部または全部が同じ計算機上で実行されていても何ら差し支えない。

【0054】図1の計算機102で実行されているアプリケーション112は、ファイル・アクセスを行う前に、アクセスしたいファイルのファイル名を指定する。本実施の形態では、この時に、統合ファイル名を指定することにより、異なる計算機103、104等の記憶装置123、124等に分散して格納されているファイルにアクセスすることができる。

【0055】図1の計算機103、104上では、それぞれファイル・サーバ113および114が実行されている。ファイル・サーバ113および114は、ネットワーク1を経由して他の計算機から送信されたファイル・アクセス要求を受信し、要求に応じて、それぞれ記憶装置123および124に格納されたファイルに対するファイル・アクセスを実行し、結果をネットワーク1を経由して要求を送信したプログラムに返す。

【0056】図1の計算機101では、統合ファイル名結合子111が実行されている。統合ファイル名結合子111は、内部データ構造を用いて、統合ファイル名ツリー2の管理と統合ファイル名ツリー2に対するファイル名の解決とを行うプログラムである。

【0057】統合ファイル名ツリー2は、概念的にはクライアントが用いる共通のファイルのファイル名をすべて含む。ここで、概念的には、ファイル名とは、ディレクトリの名称と狭義のファイルの名称を含む。しかし、統合ファイル名ツリー2を表現する内部データ構造は、必ずしも共通のファイルのファイル名をすべて含む必要がない。内部データ構造の量を節約する等の目的で、統合ファイル名ツリー2の表現に必要最小限のファイル名のみ登録する方がより望ましい。本実施の形態の統合ファイル名ツリー2の図では、太線は、登録されたファイル名、ここでは具体的には登録されたディレクトリの名称を表す。細線は、登録されていないが概念的に統合ファイル名ツリー2に含まれるファイル名、具体的には、狭義のファイルの名称を表す。

【0058】上記内部データ構造については、後に図9を用いてより詳細に説明する。統合ファイル名の解決とは、統合ファイル名から、上記統合ファイル名を有するファイル実体（複数である場合もありうる）をアクセスできるファイル・サーバがどれであるか、当該ファイル・サーバ内のどこに格納されているか、どのようなアクセス制御をかけられているかなどを示す情報として、ファイル・サーバのアドレスと当該ファイル・サーバ内のローカル・パス名とアクセス制御表などを含む情報（このような情報を「結合情報（Bind Information-BI）」と呼ぶ）を生成することである。

【0059】統合ファイル名結合子111は3つの役割を果たす。第1に、アプリケーションがある統合ファイル名を指定してファイル・アクセスを行うときに、当該統合ファイル名を解決した結果得られた結合情報を当該アプリケーションに通知する。第2に、複数のファイルが同じ統合ファイル名を有する時、もし当該複数のファイルの内容が一致でなければ、当該複数のファイルに対して一致制御を行う。第3に、新たに他のファイルシステムをその統合ファイルシステムにマウントするマウント要求が発行された時に、統合ファイル名ツリー2の内容を更新し、新しくマウントされた当該他のファイル・システムを統合ファイル名システムから利用可能にする。

【0060】前記3つの統合ファイル名結合子の役割につき、より詳細に説明する。前記統合ファイル名結合子の第1の役割では、アプリケーションがある統合ファイル名を指定してファイル・アクセスを行うときに、当該統合ファイル名を解決した結果を当該アプリケーションに通知する。

【0061】例えば図1においてアプリケーション112が統合ファイル名“//a/c/g”を有するファイルにアクセスしようとするとき、アプリケーション112は統合ファイル名結合子111に当該統合ファイル名“//a/c/g”を送信する（141）。

【0062】ここで、実際に統合ファイル名結合子111に当該統合ファイル名を送信する手続きは、アプリケーションの開発者がアプリケーションにプログラムしても、あるいはあらかじめ当該手続きをライブラリとして作成しておき、アプリケーションの開発時または実行時にアプリケーションに組み込んでもよい。後者の場合、アプリケーションの開発者は前記ライブラリのサブルーチン呼び出すだけで、当該手続きの詳細を知る必要がないという利点がある。なお、アプリケーション112が実行する141以外の手続きにおいても、これは同じである。

【0063】アプリケーション112から統合ファイル名を受信した統合ファイル名結合子111は、自分が管理している統合ファイル名ツリー2を参照し、受信した統合ファイル名を解決する。その結果結合情報をアプリケーションに返す（142）。ここで、受信した統合ファイル名を有するファイル実体が複数存在する時は、当該複数のファイル実体に関する情報をすべて含む結合情報が、アプリケーション112に返される。図1において、統合ファイル名結合子111の受信した統合ファイル名が“//a/c/g”の時は、アプリケーション112にファイル・サーバ113におけるファイル“/u/c/g”1235とファイル・サーバ114におけるファイル“/x/g”1245の両方の情報を含む結合情報が返される。

【0064】統合ファイル名結合子111からの応答1

42を受信したアプリケーション112は、受信した結合情報を参照し、アプリケーションが所望するファイル・アクセスの種類（読みだし、書き込み、ディレクトリの内容のリスト等）に応じ、適切なファイル・サーバにおけるファイル実体を選択してファイル・アクセスを実行する（143）。図1の場合のように、アプリケーション112が所望するファイル・アクセスの種類によっては、統合ファイル名結合子111から複数のファイル実体の情報を含む結合情報が返されることがある。この場合でも、アプリケーション112は、当該複数のファイル実体の一部だけを選択してファイル・アクセスを行うことができる。以上の手順により、統合ファイル名結合子111を用いて、アプリケーション112は統合ファイル名を指定してファイル・アクセスを行うことができる。

【0065】前記統合ファイル名結合子の第2の役割では、複数のファイルが同じ統合ファイル名を有する時、もし当該複数のファイルの内容が一致でなければ、当該複数のファイルに対して一致制御を行う。

【0066】例えば図1の場合では、記憶装置124にファイル名“/x/g”で格納されているファイル1245が、記憶装置123にファイル名“/u/c/g”で格納されているファイル1235の複製であり、両ファイルが同じ統合ファイル名“//a/c/g”を有する。ファイル1235がアプリケーション112によるファイル・アクセス143によって書き換えられた時、当該ファイル・アクセス143を実行したファイル・サーバ113から、一致制御要求が統合ファイル名結合子111に送信される（144）。一致制御要求144を受信した統合ファイル名結合子111は、統合ファイル名“//a/c/g”を有するファイルが複数あるかどうかを調べ、もしあれば（図1の場合はファイル1235と1245）、書き換えられたファイル1245の内容をファイル・サーバ113を経由して読み出し、ファイル・サーバ114を経由してファイル1245に書き込むことにより、ファイル1235と1245の内容を一致させる（145、146）。

【0067】本実施の形態では、ファイル1235が書き換えられたことを、一致制御要求144として統合ファイル名結合子111に通知するのは、ファイル・サーバ113が行っているが、ファイル1235への書き込みを行ったアプリケーション112が一致制御要求144と同等の通知を統合ファイル名結合子111に送信してもよい。統合ファイルの一致制御については、後に「統合ファイル名システムにおける統合ポリシー」を説明する時により詳細に説明する。

【0068】統合ファイル名結合子111の第3の役割では、新たに他のファイルシステムを統合ファイル名システムにマウントするマウント要求が発行された時に、統合ファイル名ツリー2の内容を更新し、新しくマウ

トされたファイル・システムを統合ファイル名システムから利用可能にする。

【0069】例えば図1において、記憶装置123のディレクトリ“/u”が統合ファイル名ツリー2上の統合ディレクトリ“//a”に統合マウント（131）された状態で、さらに記憶装置124のディレクトリ“/x”を統合ファイル名ツリー2上の統合ディレクトリ“//a/c”に統合マウント（133）しようとする時、当該新たな統合マウント要求を受信した統合ファイル名結合子111は、すでに統合マウントされている統合ファイル名を調べ、もし統合マウント処理133により、同じ統合ファイル名を有する複数のファイルの実体が生じるとき（図1ではファイル1235と1245および1234と1244）は、前記統合ファイル名結合子111の第2の役割にて説明したのと同様の手続きにより、ファイル1235と1245に対して一致制御を行う（145、146）。その上で、今後アプリケーション112が統合ファイル名ツリー2上の統合ディレクトリ“//a/c/g”を指定した時は、アプリケーション112に返す結合情報として、ファイル・サーバ113内のファイル1235の情報だけでなく、ファイル・サーバ114内のファイル1245の情報も含めるようにする。

【0070】（C）統合ファイル名結合子による統合ファイル名ツリーの構築

図4から図6までは、図1に示した統合ファイル名ツリー2を作成する過程を示す。

【0071】図4は、統合ファイル名ツリー2の初期状態を示す。本実施の形態では、統合ファイル名ツリー内のルートディレクトリは“//”で表す。したがって、この初期状態では統合ファイル名ツリー2には他のディレクトリがまだマウントされていないルート統合ディレクトリ“//”しかない。

【0072】図5は、図4の空状態の統合ファイル名ツリー2に対して2つのディレクトリがマウントされた後の統合ファイル名ツリー2を示す。

【0073】計算機103上で実行されるファイル・サーバ113は記憶装置123に格納されているローカル・ファイル・システムを維持している。まず、当該ローカル・ファイル・システム上のディレクトリ“/u”1231が統合ファイル名ツリー2上の統合ディレクトリ“//a”に統合マウントされる（131）。この結果、ディレクトリ//aが統合ファイル名ツリー2に登録され、ローカル・ディレクトリ“/u”1231に参照する。この結果、統合ファイル名ツリー2上でファイル・サーバ113上のディレクトリ“/u”1231とその下のディレクトリ群やファイル群をアクセスすることができる。例えば、パス“//a”は“/u”1231を、“//a/c”は“/u/c”1232を、“//a/d”は“/u/d”1233を、“//a/c/

f”は“/u/c/f”1234を、“//a/c/g”は“/u/c/g”1235をそれぞれ参照する。

【0074】計算機105上で実行されるファイル・サーバ115は記憶装置125に格納されているローカル・ファイル・システムを維持している。当該ローカル・ファイル・システム上のディレクトリ“/v”1251が統合ファイル名ツリー2上の統合ディレクトリ“//b/e”に統合マウントされる(132)と、ディレクトリ“//b/e”が統合ファイル名ツリー2に登録され、ローカル・ディレクトリ“/v”1251に参照する。このツリー2上でファイル・サーバ115上の“/v”1251とその下のディレクトリ群やファイル群をアクセスすることができる。例えば、パス“//b/e”は“/v”1251を、“//b/e/i”は“/v/i”1252を、それぞれ参照する。注意すべきところは、他のディレクトリがマウントされていないディレクトリ、例えば、ルートディレクトリ“/”とディレクトリ“//b”が存在できることである。本実施の形態ではこのようなディレクトリも、それらが統合ファイル名ツリー2上にある以上、統合ディレクトリと呼ぶ。

【0075】図6は、図5の状態からもう2つのディレクトリがマウントされた統合ファイル名ツリー2の状態を示す。

【0076】計算機104上で実行されるファイル・サーバ114は記憶装置124に格納されているローカル・ファイル・システムを維持している。当該ローカル・ファイル・システム上のディレクトリ“/x”1243を統合ファイル名ツリー2上のすでに存在する統合ディレクトリ“//a/c”に複製ディレクトリとしてマウントすることもできる(133)。このマウントの場合には、統合ディレクトリ“//a/c”とその下の統合ディレクトリ群や統合ファイル群は複製としてローカル・ディレクトリ“/x”1243に丸ごとコピーされる。図6の場合は、“//a/c/f”を“/x/f”1244に、“//a/c/g”を“/x/g”1245にコピーされる。

【0077】計算機105に新規ファイル・サーバ115'が新たに加えられたと仮定する。当該ファイル・サーバ115'は記憶装置125'に格納されているローカル・ファイル・システムを維持している。当該ローカル・ファイル・システム上のディレクトリ“/y/z”1255を統合ファイル名ツリー2上のすでに存在する統合ディレクトリ“//b/e”に追加して登録されるようにマウントすることができる(134)。ファイル・サーバ115上の“/v”1251とファイル・サーバ115'上の“/y/z”1255が統合ファイル名ツリー2上の同じ統合ディレクトリ“//b/e”に統合マウントされる結果として、ディレクトリ“//b/e”はディレクトリ“/v”1251と“/y/z”1

255とをマウントしたディレクトリとなる。従って、“//b/e”の下にある“i”と“j”は“/v”1251の下にある“i”1252と“/y/z”1255の下にある“j”1256をそれぞれ参照する。

【0078】図6には図示していないが、複数のディレクトリが統合マウントされる時に、マウント元のディレクトリに属するいずれかのファイルとマウント先のディレクトリに属する他のファイルとが同名のファイルであれば、それらの同名のファイルに対して一致制御を行う。統合ファイルの一致制御については、次の「統合ファイル名システムにおける統合ポリシー」節にてより詳細に説明する。

【0079】(D) 統合ファイル名システムにおける統合ポリシー

統合ファイル名システムにおけるディレクトリ統合を用いて、記憶装置に分散して格納されている複数のファイルの実体(以後、適宜「記憶装置に格納されたファイルの実体」を「ファイル」と称する)は一つの統合ファイル名を有することができる。ここでこれらのファイルは五種類に定義される。

【0080】(a) 基本ファイル: 統合ファイル名が参照するファイルが一つしかない場合はそのファイル、複数ある場合はマウント要求の発行者により指定されたファイルである。

【0081】(b) 複製ファイル: 基本ファイルと比べて、物理レベルの中身が同じであるファイル、即ち、完全にコピーされ、同じバイナリ・コードを持っているファイルである。

【0082】(c) 同等ファイル: 基本ファイルと比べて、物理レベルの中身が違うが論理レベルの中身が同じであるファイル、例えば、異種の編集ツールで作った同じ文章とか、異種機上でコンパイルした同じコマンドの実行コードなどである。即ち、これらのファイルは、ファイル・フォーマットが違うが、論理的な中身は同じであると考えることができる。

【0083】(d) 同類ファイル: 基本ファイルと比べて、物理レベルの中身が違うが論理レベルの中身が同じ意味を持っているファイル、例えば、違うプログラムによる大体同じ仕様を持っているプログラムとか、違う記者による同じニュースに関するレポートとか、違う言語に翻訳された文章などである。

【0084】(e) 異種ファイル: 基本ファイルと比べて、物理レベルの中身も論理レベルの中身も意味も全然違うファイル、即ち、上記の四つの種類の以外のファイルである。

【0085】一つの統合ファイル名に対して上記5つの種類が混在することができる。

【0086】統合ファイル名システムにおける統合ポリシーは、複数のファイルが一つの統合ファイル名を有する時に、どの種類のファイルが統合できるかに関する基

準とどのファイルにどのようにアクセスするかを基準を定める。例えば、次の統合ポリシーが設定できる。

【0087】(a) 基本ファイルと複製ファイルと同等ファイルしか統合できない；(b) 基本ファイルだけがあれば基本ファイルにアクセスする；(c) 複製ファイルがあれば一番早くアクセスできたファイルにアクセスする；(d) 同等ファイルがあればクライアント計算機が認識できるファイルにアクセスする；(e) 複製ファイルと同等ファイルの両方があれば(c)かつ(d)に従う。

【0088】複製ファイルのみで構成されるディレクトリを特に複製ディレクトリと呼ぶ。複製ディレクトリは、頻繁に使われるので、これを簡単に指定する方法を統合ファイル名システムは備える。即ち、あるマウント先の共通ディレクトリに統合対象となったマウント元のローカルディレクトリがマウントされる時に、統合ポリシーに当該マウント元のローカルディレクトリが複製ディレクトリであると指定されると、当該マウント元のローカルディレクトリに当該マウント先の共通ディレクトリとそのサブディレクトリよりなるサブツリーが複製される。例えば図1において、マウント要求133が発行される時に「ファイル・サーバ114上のディレクトリ“/x”1243が複製ディレクトリであり、基本ファイルを複製する」という統合ポリシーが設定されると、統合ディレクトリ“/a/c”とその下の基本ファイルは自動的にファイル・サーバ114上のディレクトリ“/x”1243に複製される。

【0089】統合ファイル名システム本実施の形態では、統合対象となったマウント元のローカルディレクトリを統合ファイル名ツリーにあるマウント・ポイントにあるマウント先の共通ディレクトリにマウントする毎に、統合ポリシーを設定することが必要となる。マウント毎の統合ポリシーの設定により、統合ファイル名ツリーの使用目的に合わせる種々のサブツリーを作ることができる。例えば、後述の読み出し専用(read-only)ポリシーを使えば、統合ファイル名ツリーに読み出し専用(read-only)サブツリーを作ることができる。後述の複製ポリシーを使えば、統合ファイル名ツリーに複製を持つサブツリーを作ることができる。

【0090】統合ポリシーはマウント先毎とマウント元毎の二つの部分に分けて設定される。統合ポリシーのマウント先毎の部分の設定は最初のマウント元をマウント先にマウントする際に行われる。統合ポリシーの内、マウント元毎の部分の設定はマウント元をマウント先にマウントする際に毎回行われる。統合ポリシーの内、マウント先のディレクトリに関する部分は、後に説明する結合情報ヘッダ413(図9)内に記憶される。統合ポリシーの内、マウント元のディレクトリに関する部分は、後に説明する結合ユニット412、412'、416または416'(図9)内に記憶される。統合ポリシーに

は次のデータ項目が含まれている。

【0091】1. マウントの許可条件

a. 統合できるファイル種類(マウント先毎)

この情報は統合ファイル名ツリーのいずれかのディレクトリがマウント先に指定されたときに、そのマウント先のディレクトリに対応して記憶される。この情報は、マウント先のディレクトリにあるファイルと同名のファイルがマウント元のディレクトリにある場合に、マウントを許されるマウント元のファイルの種類として、基本ファイル、複製ファイル、同等ファイル、同類ファイルと異種ファイルの中からいくつかを指定する。このファイル種類は、後に述べるように、結合情報ヘッダ413(図11)内の統合ポリシー表4133内に統合ファイル種類4171として記憶される。その記憶された具体的な内容の例は、図21のビットマップ571に示される。

【0092】b. 統合する方式(マウント元毎)

この情報は、統合ファイル名ツリーのいずれかのディレクトリに他のマウント元のディレクトリがマウントされたときに、そのマウント先に対応して、かつ、そのマウント元のディレクトリ別に記憶される。この情報は、マウント元のファイルとしてオリジナルか複製かを一つ選ぶ。オリジナルの場合は、マウント元におけるサブツリーに含まれたファイルの種類、即ち、基本ファイルか複製ファイルか同等ファイルか同類ファイルか異種ファイルかの一つを指定する。

【0093】統合方式として、オリジナルが指定された場合には、マウント先がマウントを許すファイルの種類の中にそのオリジナルと指定されたファイルの種類が含まれるか否かを判断され、それによりそのマウント元のファイルのマウントを許すか否かが決められる。統合方式として複製が指定された場合には、このマウント元のローカルディレクトリは複製ディレクトリとして扱われ、マウント先のディレクトリのファイルの写しがマウント元のローカルファイルに転送される。このファイル種類は、後に述べるように、結合ユニット412(図10)内に統合方式4123として記憶される。その記憶された具体的な内容の例は、図21のビットマップ572に示される。

【0094】2. アクセス制御表

a. マウント先のアクセス制御表(マウント先毎)

この表は、いずれかの統合ディレクトリに関するアクセス権限情報を保持する。すなわち、統合ファイル名ツリーのいずれかのディレクトリがマウント先に指定されたときに、そのディレクトリに関するアクセス権限情報として記憶される。この情報は、マウント先のディレクトリにおけるサブツリーにあるファイルをアクセスする権限を指定するもので、マウント先ディレクトリ権限情報と呼ぶことができる。アクセス権限情報としては種々のものを使用可能である。例えば、この情報は、ユーザ或

はグループ毎のアクセス権限項目のリストからなる。そのアクセス権限項目は、ユーザ或はグループのIDと、そのユーザ或はグループがそのディレクトリに属するファイルを書き換える権限及び読み出すことができる権限の有無のしるしからなる。このアクセス権限情報を使用するときには、統合ファイル名システムを使用するアプリケーションプログラムを起動したユーザのIDまたはそのユーザが属するグループのIDが使用される。

【0095】マウント要求は、この表の情報として特定のアクセス権限情報を設定するか、何も設定しないかを選択する。何も設定しないと全てのユーザがアクセス可になる。このアクセス制御表はマウント先ディレクトリの全サブツリーに作用する。このアクセス制御表は、後に述べるように、結合情報ヘッダ413（図11）内の統合ポリシー表4133内にアクセス制限表4172として記憶される。

【0096】b. マウント元のアクセス制御表（マウント元毎）
この情報は、統合ファイル名ツリーのいずれかのディレクトリにマウントされたマウント元のローカルディレクトリに関するアクセス権限情報である。すなわち、この情報は、統合ファイル名ツリーのいずれかのディレクトリに他のマウント元のディレクトリがマウントされたときに、そのマウント先に対応して、そのマウント元のディレクトリに関するアクセス権限情報として記憶される。このアクセス制限表は、マウント元のディレクトリにおけるサブツリーにあるファイルをアクセスする権限を指定する情報を含み、マウント元ディレクトリアクセス権限情報と呼ぶことができる。この情報の例は、上に述べて、統合ディレクトリに関するものと同じである。

【0097】マウント要求はこの表に含めるアクセス権限として特定の情報を指定するか、何も設定しないかを選択する。何も設定しないと、全てのアクセス元のユーザがアクセス可になる。このアクセス制御表はマウント元ディレクトリの全サブツリーに作用する。このアクセス制限表は、後に述べるように、結合ユニット412（図10）内にアクセス制限表4124として記憶される。

【0098】アクセス要求は、上記2種のアクセス制御表によるアクセス権限の有無を判断するための情報、たとえばユーザ或はグループのIDを指定する。そのアクセス要求が要求するファイルが属する統合ディレクトリへのアクセスが許されるか否かが、前述の統合ディレクトリに関するアクセス制御表により判断される。また、そのファイルが属するローカルディレクトリへのアクセスが許されるか否かは、前述の統合ディレクトリに対するアクセスが許された状態で、このローカルディレクトリに関するアクセス制御表により判断される。結局、このローカルディレクトリへのアクセスが許されるか否かは両方のアクセス制御表がアクセスを許すときに許され

る。このため、あるファイルまたはディレクトリに対するアクセス権限は、そのファイルまたはディレクトリにおけるマウント先とマウント元により、すなわち、そのファイルまたはディレクトリが使われた場所により決めることができる。

【0099】3. 統合ファイルの読み出し操作（マウント先毎）

この情報は、統合ファイル名ツリーのいずれかのディレクトリがマウント先に指定されたときに、そのディレクトリに対応して記憶される。この読み出し操作は、マウント先のディレクトリにあるファイルの実体を読み出すアクセス要求が発生し、アクセス対象のファイルと同名のファイルが複数あると判明した場合、それらの同名ファイルの内、読み出すべきファイルを選択する方法を指定することができる。この読み出し操作の例は次のデフォルトポリシーにおいて与えられる。この読み出し操作情報は、この操作を実現するプログラムへのポインタ（例えば、プログラム名称、アドレスなど）を保持する。この読み出し操作情報は、後に述べるように、結合情報ヘッダ413（図11）内の統合ポリシー表4133内に読み出し操作4173として記憶される。

【0100】4. 統合ファイルの書き込み操作（マウント先毎）

この情報は、統合ファイル名ツリーのいずれかのディレクトリがマウント先に指定されたときに、そのディレクトリに対応して記憶される。この書き込み操作は、マウント先のディレクトリにあるファイルの実体へ書き込みを行うアクセス要求が発生し、アクセス対象のファイル名を同名のファイルが複数あると判明した場合、それらの同名ファイルの内、書き込むべきファイルを選択する方法を指定することができる。一般には基本ファイルに対して、書き込み要求を実行するが、基本ファイル以外の同名のファイルに対しては、その内容を基本ファイルと一致させるための処理（一致制御）を行う。この書き込み操作情報の例は次のデフォルトポリシーにおいて与えられる。この書き込み操作情報は、この操作を実現するプログラムへのポインタ（例えば、プログラム名称、アドレスなど）を保持する。この情報は、後に述べるように、結合情報ヘッダ413（図11）内の統合ポリシー表4133内に書き込み操作4174として記憶される。

【0101】以下はいくつかの典型的な統合ポリシーを挙げる。

【0102】（1）デフォルト・ポリシー

1. マウントの許可条件

a. 統合できるファイル種類

基本ファイルと複製ファイルと同等ファイルしか統合できない。

【0103】b. 統合する方式

オリジナルである。ファイル種類はマウント要求の発行

者により指定される。

【0104】2. アクセス制御表

a. マウント先のアクセス制御表

何も設定しない。

【0105】b. マウント元のアクセス制御表

何も設定しない。

【0106】3. 統合ファイルの読み出し操作

a. 基本ファイルだけが存在すれば、基本ファイルを読み出す。

【0107】b. 複製ファイルが存在すれば、一番早く取れるファイルを読み出す。

【0108】c. 同等ファイルが存在すれば、クライアント計算機が認識できるファイルを読み出す。このようなファイルが複数あればその中に一番早く取れるファイルを読み出す。

【0109】d. 同類ファイルが存在すれば、ユーザが読める或は好きなファイルを読み出す。このようなファイルが複数あればその中に一番早く取れるファイルを読み出す。

【0110】e. 上記の四種類のファイルが混在すれば、上記のbかつcかつdに従う。

【0111】f. 上記の四種類のファイルが存在しなければ、「ファイルが存在しない」というエラーを返す。

【0112】4. 統合ファイルの書き込み操作

a. 基本ファイルだけが存在すれば、基本ファイルに書き込む。基本ファイル以外の同名ファイルが存在すれば、以下のようにして一致制御を実行する。

【0113】b. 複製ファイルが存在すれば、まず基本ファイルに書き込む。基本ファイルをクローズした時、更新した基本ファイルをすべての複製ファイルにコピーする。

【0114】c. 同等ファイルが存在すれば、まず基本ファイルに書き込む。基本ファイルをクローズした時、更新した基本ファイルをすべての同等ファイルに変換する。

【0115】d. 同類ファイルが存在すれば、まず基本ファイルに書き込む。基本ファイルをクローズした時、更新した基本ファイルをすべての同類ファイルに変換する。

【0116】e. 上記の四種類のファイルが混在すれば、上記のbかつcかつdに従う。

【0117】f. 上記の四種類のファイルが存在しなければ、新規基本ファイルを作る。

【0118】デフォルト・ポリシーを少し変更すれば次のようなポリシーを作ることができる。

【0119】(2) read-onlyポリシー

1. マウントの許可条件

デフォルト・ポリシーと同じである。

【0120】2. アクセス制御表

a. マウント先のアクセス制御表

read-onlyである。従って、マウント先におけるサブツリーに入った全てのディレクトリとファイルを読み出すことしかない。

【0121】b. マウント元のアクセス制御表
何も設定しない。

【0122】3. 統合ファイルの読み出し操作
デフォルト・ポリシーと同じである。

【0123】4. 統合ファイルの書き込み操作
禁止される。

【0124】(3) 複製ポリシー

1. マウントの可否判断

a. 統合できるファイル種類

基本ファイルと複製ファイルしか統合できない。

【0125】b. 統合する方式

複製である。マウント元が空であれば、マウント先に含まれた基本ファイルとサブディレクトリを全部に当該マウント元の下にそのままコピーする；空でなければ、「複製ディレクトリとして使えない」というエラーを返す。

【0126】2. アクセス制御表

マウント元およびマウント先の両方に関してデフォルト・ポリシーと同じである。

【0127】3. 統合ファイルの読み出し操作
デフォルト・ポリシーと同じである。

【0128】4. 統合ファイルの書き込み操作
デフォルト・ポリシーと同じである。

【0129】よく使われる統合ポリシーを指定する情報を含む設定ファイルを事前に用意して、マウントする時にこの設定ファイルを指定する方法を採れば、簡単に統合ポリシーを設定することができる。または、設定ファイルが用意されていない統合ポリシーを使用するときには、マウント時に必要なデータ項目をそれぞれ入力するか、或は内容が近い統合ポリシーに対して準備された設定ファイルがあるときには、その設定ファイルに対して修正、追加、削除等の編集を行うことにより、所望の統合ポリシー用の設定ファイルを簡単に作成できる。統合ポリシーを支持する各種のプログラムが必要である。これらのプログラムの追加や変更や削除により、支持され使える統合ポリシーが変わる。

【0130】(E) 複数の統合ファイル名結合子による統合ファイル名ツリーの分割管理

これまでは統合ファイル名結合子は分散計算機環境内にただ1つ存在し、当該統合ファイル名結合子が統合ファイル名ツリーを構築するとしてきたが、統合ファイル名結合子を分散計算機環境内に複数設け、1つの統合ファイル名ツリーを当該複数の統合ファイル名結合子で分担させることも可能である。1つの統合ファイル名結合子に統合ファイル名ツリー全体の管理のためのデータやアプリケーション等からの通信が集中すると、統合ファイル名結合子の実行性能が低下する。しかし、統合ファイ

ル名結合子を分散計算機環境内に複数設けると、この性能低下を防止できる。また、計算機の障害等によりいずれかの統合ファイル名結合子が正常に動作しない場合でも分散ファイルシステムとしての機能を維持し続けることができる。

【0131】図7に1つの統合ファイル名ツリーを複数の統合ファイル名結合子によって構築する場合の一例を示す。計算機201、201'、202、203、204がネットワーク1に接続されており、各計算機では統合ファイル名結合子211、211'、212、213、214がそれぞれ実行されている。ここで、統合ファイル名ツリー2が階層的に4つの領域221、222、223、224に分割され（分割された統合ファイル名ツリーの1つ1つを「領域」と呼ぶ）、それぞれの領域を1つ以上の統合ファイル名結合子が管理する。図7では、統合ファイル名結合子211および統合ファイル名結合子211'が共に領域221を、統合ファイル名結合子212が領域222を、統合ファイル名結合子213が領域223を、統合ファイル名結合子214が領域224を管理している。また、それぞれの統合ファイル名結合子は自分の管理している領域に関する情報を格納する結合情報データ構造（Bind Information DataBase-BIDB）を持っている。図7では、統合ファイル名結合子211が結合情報データ構造231を、統合ファイル名結合子211'が結合情報データ構造231'を、統合ファイル名結合子212が結合情報データ構造232を、統合ファイル名結合子213が結合情報データ構造233を、統合ファイル名結合子214が結合情報データ構造234を持っている。

【0132】図7の統合ファイル名ツリー2を利用してファイルにアクセスしようとするアプリケーション（図7には図示していない）は、まず自分の知っている統合ファイル名結合子（ここでは統合ファイル名結合子213とする）に、図1の141と同様に統合ファイル名“//a/c/g”を指定して要求を送信する。

【0133】前記要求を受信した統合ファイル名結合子213は、指定された統合ファイル名が自分の管理する領域の中にあるかどうか調べ、なければ別の統合ファイル名結合子を選び、その統合ファイル名結合子に前記アプリケーションからの要求を転送する。ここでは、アプリケーションの指定した統合ファイル名が“//a/c/g”であり、領域223外にあるので、別の統合ファイル名結合子に要求が転送される。統合ファイル名“//a/c/g”が領域223の子領域（ここで、領域224）にも入っていないため、領域223の親領域（ここで、領域221）から統合ファイル名“//a/c/g”の入っている領域を探す。要求の転送先としては、領域221を担当している統合ファイル名結合子211または統合ファイル名結合子211'のいずれか

一方が選ばれる。ここで、前記2つの統合ファイル名結合子の内どちらか選ばれるかは、過去に当該アプリケーションが利用した方を利用する方法、あるいは両方に要求を送信し、先に応答の返ってきた方を利用する方法などにより決められる。ここでは統合ファイル名結合子211が選ばれたものとする。

【0134】要求を転送された統合ファイル名結合子211も、同様に指定された統合ファイル名が自分の管理する領域221の中にあるかどうか調べる。ここでは、統合ファイル名“//a/c/g”は領域221にないが、その一部である“//a”を有する領域221の子領域222に入っている可能性がある。従って領域222を管理する統合ファイル名結合子212が選ばれる。

【0135】要求を転送された統合ファイル名結合子212も、同様に指定された統合ファイル名が自分の管理する領域222内にあるかどうか調べる。当該統合ファイル名“//a/c/g”は領域222内に存在するので、統合ファイル名結合子212は次に自分の保持しているデータを参照して、図1の142と同じく該当する統合ファイルの結合情報を前記要求を送信してきたアプリケーションに返す。

【0136】もしアプリケーションが、過去の履歴等により、自分が所望の統合ファイル名“//a/c/g”の存在する領域222を知っている場合は、上記の手順を踏まずに、直接統合ファイル名結合子212に図1の141と同様の要求を送信することができる。これにより、領域221を管理する統合ファイル名結合子211、統合ファイル名結合子211'などにかかる処理負荷をさらに低減することが可能である。

【0137】本実施の形態では、以上の手順により、統合ファイル名ツリー2を複数の統合ファイル名結合子で分割して管理させた場合でも、図1の場合と同じように統合ファイル名ツリーを利用してアプリケーションからファイルにアクセスできるようにすることが可能である。しかも、統合ファイル名ツリーを分割して複数の統合ファイル名結合子で管理することにより、管理データや通信の1つの統合ファイル名結合子への集中による統合ファイル名結合子の実効性能低下を防ぐことができる。さらに、新しく領域を追加したり、領域内のファイル名ツリーを変更したりといった操作も、当該領域を管理する統合ファイル名結合子に対してのみ行えばよいため、統合ファイル名ツリー2の構成変更も非常に容易である。

【0138】また、本実施の形態では、複数の統合ファイル名結合子が1つの領域を管理するようにすることができる。このため、当該複数の統合ファイル名結合子のどれか1つが実行されている計算機に障害が発生した場合などでも、アプリケーションから統合ファイル名ツリー2を引き続き利用することができる。

【0139】(F) 統合ファイル名結合子のデータ構造既に述べたように、統合ファイル名ツリーは、統合ファイル名結合子が保持しているデータ構造により管理されている。ここでは、当該データ構造につき、より詳細に説明する。

【0140】図8は、アプリケーション、統合ファイル名結合子、およびファイル・サーバの各プログラムの相互作用により、アプリケーションが統合ファイル名ツリーを利用してファイルにアクセスしている様子を示したものである。図8には図示していないが、これらのプログラムが動作し、また互いに通信するためのハードウェア構成は、すでに図1を用いて説明した構成と同様である。

【0141】アプリケーション312は、統合ファイル名結合子リスト320を保持している。統合ファイル名結合子リスト320は、1つ以上の統合ファイル名結合子のアドレスを格納したテーブルである。アプリケーション312は、統合ファイル名結合子に要求341を出す際に、統合ファイル名結合子リスト320に記載のアドレスに要求を送信する。統合ファイル名結合子リスト320に記載のアドレスが複数ある時は、どれか1つを選んで要求を送信してもよく、また1つ以上の統合ファイル名結合子に同時に要求を送信することもできる。要求341は、統合ファイル名と、アプリケーション312のアドレス、およびアプリケーションが所望のマウント操作或はアクセス種類（読み込みのみ、書き込みなど）からなる。

【0142】統合ファイル名結合子311は、結合情報データ構造4を保持する。結合情報データ構造は、統合ファイル名結合子内に1つずつ設けられ、当該統合ファイル名結合子の管理する領域に関する情報がすべて格納される。なお、以下説明する結合情報データ構造311の内容は、統合ファイル名ツリー2を分割して統合ファイル名結合子311とともに管理する統合ファイル名結合子311'等の結合情報データ構造でも同様である。結合情報データ構造311は、1つ以上の結合情報411、411'等と、1つ以上の結合子位置(Binder Location-BL)421、421'等を含んでいる。結合情報データ構造については、後に図9を用いてより詳細に説明する。

【0143】結合情報411は、図11を用いて後述するが、ある統合ファイル名を有するファイルの実体321、321'が格納されたファイル・サーバ313、313'のアドレスおよびファイル・サーバ内のローカル・パス名などを管理している。要求341を送信したアプリケーション312には、この結合情報411のコピーが応答として返される(342)。結合情報411は、1つ以上の結合ユニット(Bind Unit-BU)412、412'等を含んでいる。1つの結合ユニットは、マウント・ポイントに1つのディレクトリの実

体に関する情報(ファイル・サーバのアドレス、ファイル・サーバ内のローカル・パス名、アクセス制御など)を含んでいる。結合ユニットのより詳細な内容については、図10を用いて後述する。図8では、結合ユニット412が、ファイル・サーバ313のアドレスを、結合ユニット412'が、別のファイル・サーバ313'のアドレスを含んでいる。1つの結合情報が複数の結合ユニットを含んでいるときは、同じ統合ファイル名を持つディレクトリの実体が、複数存在することを意味する。なお、以上の説明は、結合情報411'等でも同様である。

【0144】結合子位置421は、要求341に含まれる統合ファイル名が、統合ファイル名結合子311の管理する領域内に存在しないときに、既に述べたように他の統合ファイル名結合子311'等に要求を転送(341')するため、統合ファイル名結合子311によって参照される(結合子位置421'等でも同様である)。結合子位置421は、ある1つの領域を管理する統合ファイル名結合子のアドレスを含んでいる。当該領域を管理する統合ファイル名結合子が(前記したように)複数ある時は、結合子位置421に複数の統合ファイル名結合子のアドレスが含まれる。図8では、結合子位置421が統合ファイル名結合子311'のアドレスを含んでおり、統合ファイル名結合子311は結合子位置421を参照して統合ファイル名結合子311'に要求を転送(341')している。要求を転送された統合ファイル名結合子311'は、統合ファイル名結合子311と同様に結合情報データ構造を保持し、その内容として結合情報と結合子位置を含んでいる。統合ファイル名結合子311'は、結合情報411'が要求341'に含まれている統合ファイル名に対応するときは、結合情報411'の内容のコピーをアプリケーション312に返す(342')。この時、要求341'に記載された、アプリケーション312のアドレスが参照される。

【0145】応答342(または342')によって、結合情報411(または411')のコピーを受信したアプリケーション312は、当該結合情報の内容を参照し、適切なファイル・サーバ(図8ではファイル・サーバ313)を選択し、当該ファイル・サーバとの間でファイル・アクセス343、344を行う。

【0146】図9は、結合情報データ構造の詳細な内容を示す。結合情報データ構造4は2つの部分からなる。第1部分は、マウント先(即ち、統合ファイル名ツリーにおけるマウント・ポイント)に関する情報を保持する結合情報411、411'等と複数の結合情報をまとめる結合情報テーブル410である。第2部分は、親領域、現在領域、子領域それぞれの統合ファイル名結合子位置情報を保持する結合子位置テーブル422、422'等と複数の結合子位置テーブルをまとめる領域テーブル420である。結合情報データ構造ヘッダ400は

結合情報テーブル410へのポインタ401と領域テーブル420へのポインタ402を保持している。

【0147】結合情報411は1つのマウント先に関する情報を保持する。結合情報411は次の情報を含む。第1は、当該マウント先の統合ファイル名4131である。第2は、マウント元に関する情報を保持する結合ユニット412、412'等と複数の結合ユニットをまとめる結合ユニットテーブル414である。第3は、当該マウント先における統合ポリシー表4133である。第4は、当該マウント先における一致制御表4134である。結合情報ヘッダ413は上記の情報の実体4131、4133、4134と情報の実体へのポインタ4132を保持する。なお、以上の説明は、結合情報411'等でも同様である。

【0148】結合ユニット412は、1つのマウント元に関する情報を保持する。この情報によってマウント元へのアクセスが可能である。図10に示すように、結合ユニット412は次の情報を含む。第1は、当該マウント元のファイル・サーバのアドレス4121である。第2は、当該マウント元の、当該ファイル・サーバ内でのローカル・パス名4122である。第3は、当該マウント元が、オリジナル・ディレクトリか複製ディレクトリかを示す統合方式4123である。第4は、当該マウント元におけるディレクトリやファイルに対するアクセス制御表4124である。第5は、当該マウント元の他の各種属性4125（サイズ、タイム・スタンプ、ファイル・サーバ種類、およびファイル種類など）である。なお、以上の説明は、結合ユニット412'等でも同様である。

【0149】上記の統合ポリシー表4133は、前記マウント先に対する次の設定を含む（図11に示す）。第1は、当該マウント先の統合ファイル種類4171である。第2は、当該マウント先における統合ディレクトリや統合ファイルに対するアクセス制御表4172である。第3は、当該マウント先における統合ファイルの読み出し操作4173である。第4は、当該マウント先における統合ファイルの書き込み操作4174である。これらの情報仕方については「統合ファイル名システムにおける統合ポリシー」節に記載した。

【0150】上記の一致制御表4134は、統合ファイル名結合子の一貫性保持機構で用いられる。当該一致制御表4134は、前記マウント先について次の情報を含む（図11に示す）。第1は、現時点での統合ファイルの一貫性の状態を示す一致状態4181である。第2は、一貫性がとられていない時間に許容範囲の最大値を示す許容不一致時間4182である。第3は、複数のマウント元のうち、一貫性がとられていないディレクトリまたはファイルの実体に関する情報を記録する結合ユニット416、416'をまとめる不一致リスト415へのポインタ4183である。

【0151】結合子位置テーブル422、422'等は、親領域、現在領域と複数の子領域に1つずつ設けられ、当該領域と当該領域に対応づけられる統合ファイル名結合子との対応関係を保持する。ここで、結合子位置テーブル422は1つ以上の結合子位置421、421'等を保持する。即ち、親領域に1つ以上の統合ファイル名結合子が対応づけられる。結合子位置421、421'等は、統合ファイル名結合子が動作する計算機のアドレスと当該計算機内での当該統合ファイル名結合子のID番号を含む。結合子位置テーブル422'等も結合子位置テーブル422と同様なデータ構造を持つ。

【0152】親領域の結合子位置テーブルまたは子領域の結合子位置テーブルをみることで、ある統合ファイル名を解決する際に次にアクセスすべき統合ファイル名結合子がわかる。アプリケーション側の計算機は、結合子位置テーブルに保持される統合ファイル名結合子のうちどれか1つに統合ファイル名の解決の要求を送ればよい。この自由度は、以下の利点がある。第1に、統合ファイル名の解決の処理コストを低減するための動的な最適化が可能である。例えば、アプリケーション側の計算機が、当該計算機からもっとも近い統合ファイル名結合子を記憶しておき、当該統合ファイル名結合子を用いることを指定することができる。第2に、1つの統合ファイル名結合子が故障によって動作しなくなった場合、別の統合ファイル名結合子に切り替えることができる。第3に、統合ファイル名の解決にかかる負荷を、複数の統合ファイル名結合子を使うことで分散することができる。なお、現在領域の結合子位置テーブルは、結合情報データ構造の複製の一貫性を保つために用いられる。

【0153】図12から図15までは、図6と図7に示した例を用いる場合、複数の領域221、222、223、224に階層的に分散された結合情報データ構造231、231'、232、233、234の具体的な構造と内容を示す。

【0154】図12は、領域名“//”である領域（即ちルート領域）221の結合情報データ構造231とその複製231'を示す。当該結合情報データ構造231と231'の結合情報データ構造ヘッダ240は結合情報テーブル2410へのポインタ241と領域テーブル2420へのポインタ242を持つ。当該領域221にマウント・ポイントが存在しないので、当該結合情報データ構造231と231'の結合情報テーブル2410は空2411である。

【0155】当該結合情報データ構造231と231'の領域テーブル2420に関しては、以下の通りである。第1に、当該領域221がルート領域なので、親領域は存在しない。従って親領域名は空2421であり、親領域の結合子位置テーブルへのポインタも空2422である。第2に、現在領域が当該領域221なので、現在領域名は当該領域221名“//”2423である。

当該領域221の統合ファイル名結合子が2つの計算機201、201'に二重化される統合ファイル名結合子211、211'なので、ポインタ2424に指された現在領域221の結合子位置テーブル243は2つの結合子位置を保持する。当該2つの結合子位置はそれぞれ統合ファイル名結合子211、211'のアドレス2430、2430'である。統合ファイル名結合子211のアドレス2430は計算機201のアドレス2431と統合ファイル名結合子211のID番号2432からなる。統合ファイル名結合子211'のアドレス2430'は計算機201'のアドレス2431'と統合ファイル名結合子211'のID番号2432'からなる。第3に、当該領域221の子領域が領域222と領域223なので、子領域名はそれぞれ“//a”245と“//b”247である。子領域222の統合ファイル名結合子212が計算機202で稼働されるので、ポインタ2426に指された当該子領域222の結合子位置テーブル244には統合ファイル名結合子212のアドレス2440が入っている。子領域223の統合ファイル名結合子213が計算機203で稼働されるので、ポインタ2428に指された当該子領域223の結合子位置テーブル245には統合ファイル名結合子213のアドレスが入っている。

【0156】図13は、領域名“//a”である領域222の結合情報データ構造232を示す。当該結合情報データ構造232の結合情報データ構造ヘッダ250は結合情報テーブル2510のポインタ251と領域テーブル2520へのポインタ252を持つ。当該領域222にマウント・ポイントが“//a”と“//a/c”なので、当該結合情報データ構造232の結合情報テーブル2510にはマウント先“//a”の結合情報253へのポインタ2511とマウント先“//a/c”の結合情報254Bへのポインタ2512が入っている。

【0157】マウント先“//a”の結合情報253の結合情報ヘッダ2530は、当該結合情報253の統合ファイル名“//a”2531、結合ユニットテーブル2535へのポインタ2532、統合ポリシー表2533と一致制御表2534からなる。ファイル・サーバ113におけるローカル・ディレクトリ“/u”1231が統合ファイル名“//a”に統合マウントされたので、当該結合情報253の結合ユニットテーブル2535にはマウント元“/u”1231の結合ユニット2550へのポインタ2536が入っている。マウント元“/u”1231の結合ユニット2550は、ファイル・サーバ113のアドレス2551、ローカル・パス名“/u”2552、「オリジナル」である統合方式2553、マウント元“/u”1231のためのアクセス制御2554と各種属性2555を保持する。

【0158】マウント先“//a/c”の結合情報25

4の結合情報ヘッダ2540は、当該結合情報254の統合ファイル名“//a/c”2541、結合ユニットテーブル2545へのポインタ2542、統合ポリシー表2543と一致制御表2544からなる。ファイル・サーバ113におけるローカル・ディレクトリ“/u/c”1232が「オリジナル」として、ファイル・サーバ114におけるローカル・ディレクトリ“/x”1243が「複製」として、統合ファイル名“//a/c”に統合マウントされたので、当該結合情報254の結合ユニットテーブル2545にはマウント元“/u/c”1232の結合ユニット2560へのポインタ2546とマウント元“/x”1243の結合ユニット2570へのポインタ2547が入っている。マウント元“/u/c”1232の結合ユニット2560は、ファイル・サーバ113のアドレス2561、ローカル・パス名“/u/c”2562、「オリジナル」である統合方式2563、マウント元“/u/c”1232のためのアクセス制御2564と各種属性2565を保持する。マウント元“/x”1243の結合ユニット2570は、ファイル・サーバ114のアドレス2571、ローカル・パス名“/x”2572、「複製」である統合方式2573、マウント元“/x”1243のためのアクセス制御2574と各種属性2575を保持する。

【0159】複製ディレクトリが存在する場合として、マウント先“//a/c”の結合情報254にある一致制御表2544は以下の通りである。第1に、現在の一致状態は「一致」25441である。第2に、許容不一致時間は10秒25442である。第3に、現在の一致状態が「一致」であるから、不一致リストが存在しない。不一致リストへのポインタは空25443である。

【0160】結合情報データ構造232の領域テーブル252に関しては、以下の通りである。第1に、親領域がルート領域221なので、親領域名は“//”2521である。親領域221の統合ファイル名結合子が2つの計算機201、201'に二重化される統合ファイル名結合子211、211'なので、ポインタ2522に指された親領域221の結合子位置テーブル258は2つの結合子位置を持つ。当該2つの結合子位置はそれぞれ統合ファイル名結合子211、211'のアドレス2580、2580'である。第2に、現在領域が当該領域222なので、現在領域名は当該領域222名“//a”2523である。現在領域222の統合ファイル名結合子212が計算機202で稼働されるので、ポインタ2524に指された現在領域222の結合子位置テーブル259には統合ファイル名結合子212のアドレス2590が入っている。第3に、当該領域222の子領域が存在しないので、子領域名は空2525であり、子領域の結合子位置テーブルへのポインタも空2526である。

【0161】図14は、領域名“//b”である領域2

23の結合情報データ構造233を示す。当該結合情報データ構造233の結合情報データ構造ヘッダ260は結合情報テーブル2610へのポインタ261と領域テーブル2620へのポインタ262を持つ。当該領域223にマウント・ポイントが存在しないので、当該結合情報データ構造233の結合情報テーブル2610は空2611である。

【0162】結合情報データ構造233の領域テーブル2620に関しては、以下の通りである。第1に、親領域がルート領域221なので、親領域名は“//”2621である。親領域221の統合ファイル名結合子が2つの計算機201、201'に二重化される統合ファイル名結合子212、212A'なので、ポインタ2622に指された親領域221の結合子位置テーブル263は2つの結合子位置を持つ。当該2つの結合子位置はそれぞれ統合ファイル名結合子211、211'のアドレス2630、2630'である。第2に、現在領域が当該領域223なので、現在領域名は当該領域223名“//b”2623である。現在領域223の統合ファイル名結合子213が計算機130Cで稼働されるので、ポインタ2624に指された現在領域223の結合子位置テーブル264には統合ファイル名結合子213のアドレス2640が入っている。第3に、当該領域223の子領域が領域224なので、子領域名は“//b/e”2625である。子領域224の統合ファイル名結合子214が計算機204で稼働されるので、ポインタ2626に指された子領域224の結合子位置テーブル265には統合ファイル名結合子214のアドレス2650が入っている。

【0163】図15は、領域名“//b/e”である領域224の結合情報データ構造234を示す。当該結合情報データ構造234の結合情報データ構造ヘッダ270はBNIテーブル2710へのポインタ271と領域テーブル2720へのポインタ272を持つ。当該領域224にマウント・ポイントが“//b/e”なので、当該結合情報データ構造234の結合情報テーブル2710にはマウント先“//b/e”の結合情報273へのポインタ2711が入っている。

【0164】マウント先“//b/e”の結合情報273の結合情報ヘッダ2730は、当該結合情報273の統合ファイル名“//b/e”2731、結合ユニットテーブル2735へのポインタ2732、統合ポリシー表2733と一致制御表2734からなる。ファイル・サーバ115におけるローカル・ディレクトリ“/v”1251とファイル・サーバ115'におけるローカル・ディレクトリ“/y/z”1255とも「オリジナル」として、マウント先“//b/e”に統合マウントされたので、当該結合情報273の結合ユニットテーブル2735にはマウント元“/v”1251の結合ユニット2740へのポインタ2736とマウント元“/y

/z”1255の結合ユニット2750へのポインタ2737が入っている。マウント元“/v”1251の結合ユニット2740は、ファイル・サーバ115のアドレス2741、ローカル・パス名“/v”2742、「オリジナル」である統合方式2743、マウント元“/v”1251のためのアクセス制御2744と各種属性2745を保持する。マウント元“/y/z”1255の結合ユニット2750は、ファイル・サーバ115'のアドレス2751、ローカル・パス名“/y/z”2752、「オリジナル」である統合方式2753、マウント元“/y/z”1255のためのアクセス制御2754と各種属性2755を保持する。

【0165】結合情報データ構造234の領域テーブル272に関しては、以下の通りである。第1に、親領域が領域223なので、親領域名は“//b”2721である。親領域223の統合ファイル名結合子213が計算機203で稼働されるので、ポインタ2722に指された親領域223の結合子位置テーブル276には統合ファイル名結合子213のアドレス2760が入っている。第2に、現在領域が当該領域224なので、現在領域名は“//b/e”2723である。現在領域224の統合ファイル名結合子214が計算機204で稼働されるので、ポインタ2724に指された現在領域224の結合子位置テーブル277には統合ファイル名結合子214のアドレス2770が入っている。第3に、当該領域224の子領域が存在しないので、子領域名は空2725であり、子領域の結合子位置テーブルへのポインタも空2726である。

【0166】(G) 統合ファイル名ツリーにアクセス時の統合ファイル名システムの内部動作

図16のフローチャートを用いて、図8を参照しながら、エンド・ユーザがアプリケーション・プログラムを使って、統合ファイル名ツリーにアクセスする時の統合ファイル名システムの内部動作を説明する。

【0167】第1のフェーズ(アプリケーション側)：ユーザ操作によりアプリケーション312が、ディレクトリまたはファイルへのアクセスを要求した際、アクセス要求が指定した要求ファイル名を取り出し(511)、統合ファイル名であるか、即ち要求ファイル名が“//”で始まるか否かを検査する(512)。統合ファイル名であれば、この要求341は統合ファイル名ツリーに向けたものである。当該アプリケーション312は、統合ファイル名結合子リスト320に記載した適当な統合ファイル名結合子311に当該要求341を送信して(514)から、統合ファイル名結合子からの応答を待つ(515)。当該要求341には次のデータが入っている：統合ファイル名、アクセス種類と当該アプリケーション312に戻るアドレス。統合ファイル名でなければ、当該ディレクトリまたはファイルは当該アプリケーション312の動作する計算機上のファイルで

あるので、当該アプリケーション312はそれ自身が動作する計算機上のローカル・ファイル・システムにアクセスする(513)。

【0168】統合ファイル名結合子リストに記載の統合ファイル名結合子に要求を送信するステップ514では、統合ファイル名結合子リスト320から何らかのアルゴリズムを用いて統合ファイル名結合子が選択される。もし統合ファイル名結合子リスト320が空なら、エラー「統合ファイル名結合子が指定されていない」が返される、そうでなければ、上記要求341は候補である統合ファイル名結合子に送信される。もし候補である統合ファイル名結合子への送信に失敗し、失敗理由がネットワークまたは候補である統合ファイル名結合子の故障であるなら、統合ファイル名結合子リスト320上、次の統合ファイル名結合子があるかどうか調べる。ここでもし、次の統合ファイル名結合子が存在すれば、当該次の統合ファイル名結合子を候補である統合ファイル名結合子とし、上記要求341は候補である統合ファイル名結合子に送信される。このように、送信成功または統合ファイル名結合子リスト320の中のすべての統合ファイル名結合子に送信失敗まで繰り返す。失敗した場合、エラー「すべての統合ファイル名結合子が使用できない」が返される。

【0169】第2のフェーズ(統合ファイル名結合子側)：上記統合ファイル名結合子311は、アプリケーションからの要求を待つ(531)状態で、上記アプリケーション312からの要求341を受けると、当該要求341により指定された統合ファイル名とその統合ファイル名結合子311自身が動作する計算機上の結合情報データ構造の中にある領域テーブルの現在領域名とを比較(532)することにより、当該統合ファイル名が現在領域内であるかどうかを検査する(533)。

【0170】当該統合ファイル名が現在領域にあれば、当該統合ファイル名結合子311は結合情報テーブルから、当該統合ファイル名に該当する結合情報411を検索する(534)。検索結果(即ち該当する結合情報411の内容)が当該要求元(即ち上記アプリケーション312)に応答342として送られる(536)。検索結果としては、もし当該統合ファイル名に該当する結合情報411が存在すれば当該結合情報411、そうでなければエラー「ディレクトリまたはファイルが存在しない」である。もし当該統合ファイル名が、現在領域の外であれば、当該要求341は別の領域の統合ファイル名結合子に転送される(535)。

【0171】別の領域の統合ファイル名結合子へ要求を転送するステップ535の手順は、以下の通りである(図17に示す)。現在の統合ファイル名結合子311は、自分の結合情報データ構造の領域テーブルに格納された親領域名または子領域名の内、当該統合ファイル名と最も近い領域を見出し(541)、この見出された親

領域または子領域を選択された領域とする。次に、当該選択された領域の結合子位置テーブルを検索し、該当する統合ファイル名結合子を見出す(542)。該当する統合ファイル名結合子が見つかったかどうかによって、処理が分けられる(543)。もし1つ以上の統合ファイル名結合子が当該選択された領域に対応づけられている場合、適当なアルゴリズムで統合ファイル名結合子を1つ選択する。当該要求341は、その統合ファイル名結合子411'に転送される(545)。要求転送が成功したかどうかを検査され(546)、もしこの転送が失敗した場合、上記のステップ(542)に戻って、該当する新しい統合ファイル名結合子を見出す。該当する統合ファイル名結合子が見つからなかったまたはすべての該当する統合ファイル名結合子に要求が転送され、失敗した場合、エラー「統合ファイル名結合子が利用不能」が上記アプリケーション312に応答342として送られる(544)。

【0172】現在の統合ファイル名結合子311は、一連の処理を完了し、次の要求を待つ(531)。

【0173】第3のフェーズ(アプリケーション側)：上記アプリケーション312は、統合ファイル名結合子からの応答を待つ状態で上記応答342を受け取る(515)。もし結果が、上記要求341の統合ファイル名の結合情報であれば、当該アプリケーション312は、統合ファイル名結合子リスト320を更新し(518)、そして、受信した結合情報からファイル・サーバを決定し、アクセス要求343を送信する(519)。そうでなければ、即ちエラーである場合(516)、統合ファイル名結合子からのエラーが返される(517)。

【0174】統合ファイル名結合子リストを更新するステップ518は以下の手順で行なわれる。統合ファイル名結合子リスト320が検査され、上記応答342を返す統合ファイル名結合子が含まれていなければ、その統合ファイル名結合子を当該統合ファイル名結合子リスト320に追加する。そうでなければ、その統合ファイル名結合子が当該統合ファイル名結合子リスト320に既に存在するので、その統合ファイル名結合子の優先度を適当に高める。

【0175】受信した結合情報からファイル・サーバを決定し、アクセス要求送信するステップ519は以下の方法で行われる。当該受信した結合情報の統合ポリシー表に記入されたポリシーとその結合情報に入っているマウント元ごとの結合ユニットの情報(図10に示すような統合方式4123、アクセス制御表4124、各種属性4125)とを用いて、さらにアクセス要求が要求するアクセスの種類(ファイルの読み出し、ファイルの書き込み、ディレクトリの内容のリストなど)に依存して、アクセス要求送信先のファイル・サーバを決める。

【0176】すなわち、アクセス要求が指定するディレ

クトリに対応して記憶された、そのアクセス要求が指定するファイル名のファイルに対する結合情報411をアクセスする。その中の結合情報ヘッダ413に含まれた統合ポリシー表4133(図11)内に含まれている読み出し操作4173、書き込み操作4174のいずれかを、アクセス要求が要求するアクセスの種類に応じて読み出す。さらにその結合情報411に含まれた結合ユニット412、412'を読み出す。複数の結合ユニットが含まれているときには、全ての結合ユニットを読み出す。アクセス要求が指定するファイル名のファイルに対する目的の結合ユニットが複数検出されたときには、それぞれの結合ユニット内の各種属性4125の内、ファイルの種類を読み出し、そのファイルが同名ファイルであるか否か、同名ファイルであるときには同名ファイルの種別を判断する。

【0177】複数の目的の結合ユニットが検出され、それぞれの結合ユニットに記憶されたファイルの種類が同名ファイルであるときには、それらの同名ファイルのそれぞれに対してアクセスを実行するか否かと、アクセスを実行するときにはアクセス動作の内容とを、統合ポリシー表4133(図11)から読み出された読み出し操作4173または書き込み操作4174のいずれかとアクセス要求が指定するアクセスの種類に基づいて決定する。アクセスすると決定されたファイルが属するサーバ計算機に、そのファイルに対して決定されたアクセス動作を指定するアクセス要求を送信する。

【0178】なお、アクセス要求送信する前に、すでに統合ポリシーの説明の中で説明したように、マウント先毎(即ち結合情報毎)のアクセス制御表4172とマウント元毎(即ち結合ユニット毎)のアクセス制御表4124の両方のそれぞれによって同時にアクセスが許可されているかにより、当該アクセスの可否を判断する。可の場合、送信する;否の場合、拒否する。次は例を用いて三つのアクセスの種類について上記アクセス要求送信ステップ519の処理の詳細を説明する。

【0179】第1のアクセスの種類は、ファイルの読み出しである。例えば、アプリケーションは図6に示す統合ファイル“//b/e/j”を読み出そうとする。この場合、当該アプリケーションは統合ファイル名結合子から図15に示す結合情報273を受信する。結合情報273に入っている結合ユニットはファイル・サーバ115における「オリジナル」としてのマウント元“/v”1251の結合ユニット2740と、ファイル・サーバ115'における「オリジナル」としてのマウント元“/y/z”1255の結合ユニット2750である。

【0180】ここで、結合情報273における統合ポリシー表に設定されたポリシーはデフォルト・ポリシーであるとする。このポリシーにより指定される読み出し操作4173は、すでにデフォルト・ポリシーの説明の中

に記載したように、アクセス要求されたファイルに対して基本ファイル以外の同名ファイルがあれば、その同名ファイルの種類が、複製ファイル、同等ファイル、同類ファイルのいずれかであるかにより、読み出すべきファイルを決める。

【0181】今の場合には、図6の例では、ファイル名“//b/e/j”のファイルが登録されているディレクトリeには基本ファイルi、jのみが登録されていると仮定している。したがって、ディレクトリeからは、ファイルi、jのみが検出され、ファイルjの同名ファイルが検出されないで、ファイルjの基本ファイルのみがアクセスされる。その方法は以下の通りである。

【0182】もし、当該アプリケーションの所在する計算機には当該両マウント元1251、1255へのアクセス記録がなければ、統合ファイル“//b/e/j”の実体はどのファイル・サーバに格納されているかわからないので、通常両ファイル・サーバ115、115'にアクセスする必要がある。このため、ステップ519では両ファイル・サーバ115、115'を決定し、同時にアクセス要求を送信する。もし、当該両マウント元1251、1255へのアクセス記録があれば、例えば統合ディレクトリ“//b/e”の内容をリストしてから当該統合ファイル“//b/e/j”をアクセスする場合、ステップ519ではそのアクセス記録によりファイル・サーバ115'を決定することができる。

【0183】なお、図6と異なり、一般的にはディレクトリeにファイルjの同名ファイルが登録されている可能性がある。したがって、上記読み出し要求の処理は一般的には以下のようにして行われる。まず、ディレクトリeに対応して記憶された、読み出し要求が指定するファイル名に対応する複数の結合ユニットが読み出され、それぞれの結合ユニットに含まれたファイル種類が判断される。こうして、複数の同名ファイルの種類が判断されると、読み出し操作情報4173にしたがって一つの同名ファイルを選択し、そのファイルの実体を管理するサーバ計算機に読み出し要求を送る。

【0184】第2のアクセスの種類は、ファイルの書き込みである。例えば、アプリケーションは図6に示す統合ファイル“//a/c/f”を書き込もうとする。この場合、当該アプリケーションは統合ファイル名結合子から図13に示す結合情報254を受信する。結合情報254に入っている結合ユニットはファイル・サーバ113における「オリジナル」としてのマウント元“/u/c”1232の結合ユニット2560と、ファイル・サーバ114における「複製」としてのマウント元“/x”1243の結合ユニット2570である。

【0185】ここで、結合情報254における統合ポリシー表に設定されたポリシーはデフォルト・ポリシーであるとする。このポリシーにより指定される書き込み操作4174は、すでにデフォルト・ポリシーの説明の中

に記載したように、アクセス要求されたファイルに対して基本ファイル以外の同名ファイルがあれば、その同名ファイルに対しても一致制御のためのアクセスを行う。そのアクセスの種類は同名ファイルの種類が、複製ファイル、同等ファイル、同類ファイルのいずれかであるかにより異なる。その同名ファイルが複製ファイルの時には、基本ファイルへの書き込みを行い、基本ファイルがクローズされた時点で、複製ファイルに基本ファイルを複製ファイルにコピーする。

【0186】今の例では、ディレクトリcには、基本ファイル1234と複製ファイル1244が登録されている。この場合、ステップ519により決定するアクセス要求送信先のファイル・サーバは、基本ファイル1234を管理するファイル・サーバ113である。デフォルト・ポリシーに記述されたように、基本ファイル“/u/c/f”1234をクローズした時、更新した当該基本ファイルは統合ファイル名結合子の一致制御機構により複製ファイル“/x/f”1244にコピーされる。

【0187】第3のアクセスの種類は、ディレクトリの内容のリストである。例えば、アプリケーションは図6に示す統合ディレクトリ“//a/e”の内容をリストしようとする。第1のアクセスの種類と同じように、当該アプリケーションは統合ファイル名結合子から図15に示す結合情報273を受信する。ここで、結合情報273における統合ポリシー表に設定されたポリシーもデフォルト・ポリシーであるとする。リスト処理の場合には、この統合ポリシー内の読み出し操作情報を使用される。統合ディレクトリの内容をすべてリストするため、すべての「オリジナル」としてのマウント元にアクセスする必要がある。このため、ステップ519では両ファイル・サーバ115、115'を決定し、同時に両マウント元“/v”1251と“/y/z”1255のアクセス要求を送信する。

【0188】第4のフェーズ（ファイル・サーバ側）：ファイル・サーバ313は上記アクセス要求343を受け取り、当該アクセス要求343で指定されたディレクトリまたはファイルにアクセスする。そして、上記アプリケーション312にアクセス結果344を返す。

【0189】第5のフェーズ（アプリケーション側）：上記アプリケーション312は、ファイル・サーバからの応答を待つ状態で上記ファイル・サーバ313からの上記アクセス結果344を受け取る（520）。当該アプリケーション312は、当該アクセス結果344に対して適当な処理（例えば、複数のファイル・サーバに分散され、格納されたディレクトリのディレクトリ統合）を行ない（521）、処理した結果を返す（522）。

【0190】上記のアクセス内部動作は、アクセス先がマウント・ポイントであるかどうかに関わらず、統合ファイル名ツリーにあるすべての統合ファイルまたは統合ディレクトリに適用する。このため、あるマウント・ポ

イントにおける統合ポリシーはそのマウント・ポイントにおけるサブツリーにあるすべての統合ファイルまたは統合ディレクトリに適用する。すなわち、ディレクトリ統合は同名サブディレクトリにも再帰的に行われる。

【0191】（H）統合ファイル名ツリーにマウント時の統合ファイル名システムの内部動作

マウントは、システム管理者がシステム管理ツールを使用してマウント要求を統合ファイルシステムに対して発行したときに行われる。マウントに使用する統合ポリシー、マウント元ディレクトリ、マウント先ディレクトリ等の情報はこのときにシステム管理者が指定する。統合マウント方法を用いて統合ファイル名ツリーにマウントする時には、結合情報と結合ユニットのレコードが新規に作られる。即ち、新規のマウント先にマウントする時に新規に結合情報レコードと結合ユニットレコードが作られ、既に存在するマウント先に新規にマウント元をマウントする時に新規に結合ユニットレコードだけが作られる。従って、マウント要求を発行する時に結合情報と結合ユニット構造に含まれるべきデータの内、自動的に生成できないデータの入力が必要である。具体的に、結合情報レコードに必要なデータは、統合ファイル名、統合ポリシーのマウント先毎の部分（即ち、統合ファイル種類、アクセス制御表、読み出し操作、書き込み操作）と一致制御規制（即ち、許容不一致時間）であり、結合ユニットレコードに必要なデータはファイル・サーバのアドレス、ローカル・パス名と統合ポリシーのマウント元毎の部分（即ち、統合方式、アクセス制御表）と、マウント元の属性の内のファイル種類等である。

【0192】図20のフローチャートは統合ファイル名ツリーにマウント時の統合ファイル名システムの内部動作を示す。以下は、図20を用いて、図8を参照しながら、統合ファイル名システムのマウント内部動作を説明する。

【0193】第1のフェーズ（アプリケーション側）：アプリケーション312が、ディレクトリまたはファイルをマウントする要求を出した際、マウント先のファイル名を取り出し（551）、統合ファイル名であるか、即ち要求ファイル名が“//”で始まるか否かを検査する（552）。統合ファイル名であれば、この要求341は統合ファイル名ツリーに向けたものであるので、当該アプリケーション312は、統合ファイル名結合子リスト320に記載した適当な統合ファイル名結合子311に当該要求341を送信して（554）から、統合ファイル名結合子からの応答を待つ（555）。当該要求341には次のデータが入っている：前述した統合ファイル名などの結合情報レコードに必要なデータと当該アプリケーション312に戻るアドレス。統合ファイル名でなければ、当該マウント先は当該アプリケーション312が動作する計算機上のローカル・ファイル名ツリーにあるので、当該アプリケーション312は、既存のマ

ウント方法を用いて、自身が動作する計算機上のローカル・ファイル・システムにマウントする(553)。

【0194】第2のフェーズ(結合ファイル名結合子側)：上記結合ファイル名結合子311は、アプリケーションからの要求を待つ(561)状態で、上記アプリケーション312からの要求341を受けると、図16に示すように、結合情報データ構造から当該要求341にある統合ファイル名を探す(562)。これによって、上記マウント先は新規であるかどうかを判断する(563)。新規なマウント先であれば、要求341から受け取った入力データに基づいて新規結合情報とその新規結合ユニットを作成して(564、566)、当該新規結合情報の内容を上記アプリケーション312に回答342として送信する(467)。既に存在していれば、当該マウント先の結合情報に格納された結合情報毎の統合ポリシーと要求341から受け取った結合ユニット毎の統合ポリシーに基づいて当該マウントの可否を判断する(565)。マウント可であれば、当該マウント先の結合情報の中の新規結合ユニットを作成して(566)、当該結合情報の内容を上記アプリケーション312に回答342として送信する(467)。マウント否であれば、エラー「マウント拒否」を上記アプリケーション312に回答342として送信する(468)。

【0195】マウント可否の判断の仕組みは図21に示される。結合情報411毎の統合ポリシー表4133に記載の統合ファイル種類4171(図11)は、統合可能なファイル種類に対応するビットからなるマウント可否ビットマップ571として記憶されている。このビットマップ571のいずれかのビットに「1」が立っていれば、マウント元のファイルがオリジナルである場合には、該当するファイル種類のファイルのマウントが許可される。そうでなければ(即ち、「0」である場合)、該当するファイル種類のファイルのマウントが拒否される。図21の場合は、基本ファイル、複製ファイルと同等ファイルに対してマウントが許可される。それ以外のファイル種類(即ち、同類ファイルと異種ファイル)は拒否される。

【0196】結合ユニット412(図10)内の統合方式4123と各種属性4125内のファイル種類は、統合方式4123がオリジナルを示す時の複数のファイル種類に対応する複数のビットとその統合方式が複製であることにそれぞれ対応する1ビットからなるマウント方式ビットマップ572として実際には記憶されている。「1」が立っているビットは実行すべきマウント方式を示すことになる。マウント方式は六つある。一回のマウント時点で使えるマウント方式は一つしかないから、このビットマップ572に一つの「1」しか立っていない。図21の場合は、マウント方式はオリジナルの複製である。即ち、マウント時に同名になるファイルは複製

ファイルと見なす。

【0197】マウントの可否の判断のために各々のマウント方式に対応する判断プログラムが使用される。判断プログラム・テーブル576は各々のマウント方式に適用する判断プログラムのアドレスを記憶している。

【0198】マウント可否を判断する処理は二つのステップに分けられる。第1ステップでは、まず、結合情報毎の統合ポリシーのマウント可否ビットマップ571と結合ユニット毎の統合ポリシーのマウント方式ビットマップ572の論理積を取る。すなわち、図21が示すように、ビットマップ571の各ビットと、ビットマップ572内のファイルの種類を示す対応するビットとの論理積をそれぞれ取り(573)、さらに、ビットマップ571内の複製ファイルに対応するビットと、ビットマップ572内の統合方式がビットマップであるか否かを示すビットとの論理積を取り(573')、これら二つの論理積により得られる複数のビットからなるビットマップ574内の一つのビットに「1」が立っていれば、第2ステップに行く(575の「yes」)。そうでなければ、マウントを拒否する(575の「no」)。第2ステップでは、判断プログラム・テーブル576から許可されたマウント方式に適用する判断プログラムのアドレスを取り出し、結合ユニットに入ったすべての同名になるファイルに対して該当する判断プログラムを実行する(577)。図21の場合は、該当判断プログラムは複製ファイルの判断ツールであり、このプログラムは、例えば、UNIXのcmpとdiffコマンドよりなる。判断プログラムにより、マウント可否を判定する(578)。

【0199】各判断プログラムは、対応するマウント方式により定まる処理を実行する。より具体的には、結果ビットマップ574内の統合方式がオリジナルでファイル種類が基本ファイルであることに対応するビットが1のときには、同名基本ファイル存在判断ツールが実行される。このプログラムは、マウント元の基本ファイルと同名の基本ファイルがマウント先のディレクトリに存在するか否かを判定する。もし、同名の基本ファイルが存在すれば、この判断プログラムは、マウント元の基本ファイルのマウントはできないと判断する。

【0200】結果ビットマップ574内の統合方式がオリジナルで、ファイル種類がそれぞれ複製ファイル、同等ファイル、同類ファイル、異種ファイルに対応するビットがそれぞれ1のときには、それぞれ複製ファイルの判断ツール、同等ファイルの判断ツール、同類ファイルの判断ツール、異種ファイルの判断ツールが実行される。それぞれのプログラムは、マウント元のファイルと同じファイル名を有するマウント先のファイルが、それぞれ複製ファイル、同等ファイル、同類ファイル、異種ファイルであるかを判断する。もし、マウント先の同名ファイルが、その特定の種類の同名ファイルでないとき

には、それぞれの判断プログラムは、マウント元の同名ファイルのマウントはできないと判断する。

【0201】結果ビットマップ574内の統合方式が複製であることに対応するビットが1のときに実行される判断ツールは、マウント先の全ツリーをマウント元にコピーし、コピーが成功したか否かを判断するプログラムである。もし、マウント元のファイルを記憶するローカルな記憶装置の容量が不足等の理由によりコピーを行えないとき、あるいはコピーを実行中に何らかの理由にコピー動作が中断されたときには、このプログラムは、マウント元のディレクトリのマウントはできないと判断する。

【0202】第3のフェーズ（アプリケーション側）：上記アプリケーション312は、統合ファイル名結合子からの応答を待つ状態で上記応答342を受け取る（555）。もし結果が、上記要求341の統合ファイル名の結合情報であれば、当該アプリケーション312は、統合ファイル名結合子リスト320を更新し（558）、そして、当該結合情報を返して終了する（559）。そうでなければ、即ちエラーである場合、統合ファイル名結合子からのエラーが返される（557）。

【0203】今まで説明した統合マウント方法はファイルに対しても適用できる。即ち、マウント元がファイルであれば、マウント先がファイルである。マウント元としてのファイルが複数がある場合は、ディレクトリ統合による同名ファイルの処理と同じように、マウント先で当該複数のファイルをマウントする（即ち、ファイル統合）。

【0204】（1）統合ファイル名システムにおける統合ファイルの一致制御

統合ファイルの一致制御は結合情報の一致制御表と統合ポリシー表を参照する。ある結合情報レコードの一致制御表にある一致状態が「不一致」になると、この結合情報レコードに対する一致制御が起動される。統合ポリシーの書き込み操作に従って一致制御が行われ、許容不一致時間以内に「一致」状態にさせる。

【0205】統合ファイルの一致制御は、複製ファイルに対する一致制御だけではなく、同等ファイルと同類ファイルに対する一致制御も含まれている。例えば、上記のデフォルト・ポリシーの「4. 統合ファイルの書き込み操作」は、統合ファイルの一致制御の一方法を定める。そこで、複製ファイルを更新する時に簡単なコピーで済むことができるが、他の種類のファイルを更新する場合は対応する変換ツールが必要である。

【0206】（J）応用例

これまで説明してきた統合ファイル名システムは、ローカル・エリア・ネットワーク（LAN）から広域ネットワーク（WAN）まで幅広い分散計算機環境に適用でき、また前記分散計算機環境を構成する計算機として、パーソナル・コンピュータ（PC）やワークステーショ

ン（WS）から大型計算機（並列計算機を含む）まで多様な計算機を用いることができる。下記に統合ファイル名システムの応用例として、2種類の典型的な分散計算機環境を挙げ、それぞれにどのように統合ファイル名システムを適用するかを説明し、それぞれどのような効果があるかを明らかにする。

【0207】（応用例1）図18に本応用例における分散計算機環境の構成を示す。LAN600に、1台以上のデスクトップ機（PCまたはWS）601、601'等および602、602'等が接続されている。また、LAN600には、大型計算機603が接続されている。デスクトップ機601、601'等および大型計算機603には、ファイル621、621'等が格納されている。

【0208】図18の分散計算機環境において、ユーザがデスクトップ機601、601'等および602、602'等上で実行しているアプリケーションがファイル621、621'等にアクセスしようとする場合、従来の技術では、次のような問題があった。第1は、デスクトップ機601等におけるユーザ（またはユーザが実行しているアプリケーション、或はアプリケーションが使用しているファイル・システム）は、必要なファイルがどの計算機上にあるかを意識していなくてはいけなかった。第2は、ユーザがファイルを複製する時や、プログラムのファイルをバージョン・アップしたり異種機に適用する時、それらのファイルは互いに別のファイル名を持つため、すべてのファイル名を把握していなくてはならず、しかも複製ファイルのどれか1つを更新したときはその他の複製ファイルをすべて更新し、最新かつ特定計算機で実行できるプログラムを選択する必要があるなど、ユーザによる管理が困難であった。

【0209】図18に示す分散計算機環境では、例えば統合ファイル名システムを次のように適用できる。図18において、ファイル621、621'等を格納しているデスクトップ機602、602'等および大型計算機603のそれぞれにおいて、ファイル・サーバ613、613'等を実行する。さらに、大型計算機603上で、統合ファイル名結合子611を実行する。統合ファイル名結合子611が含まれている結合情報データ構造には、すでに図9ないし図15を用いて説明した方法により、ファイル・サーバ613、613'等への結合情報がすべて統合ファイル名ツリー2に基づいて保持される。デスクトップ機601、601'等および602、602'等の上のアプリケーション612、612'、612''等は、“//”で始まるファイル名を指定してファイル・アクセス要求を行うことにより、図8と図16を用いて説明した通り、統合ファイル名ツリー2に基づく統合ファイル名を用いて、ファイル621、621'等にアクセスすることができる。

【0210】上記のように統合ファイル名システムを適

用することにより、前記従来技術における問題点を解決することができる。第1に、前記アプリケーション612、612'、612"等は、複数の計算機に分散しているファイル621、621'等に、統合ファイル名ツリー2に基づく統合ファイル名を指定することでアクセスすることができる。ここで、あるファイルがどの計算機に格納されているかは統合ファイル名結合子611が決定するため、統合ファイル名の指定において、ファイルの格納された計算機を指定する必要がない。従って、ユーザ（およびユーザが実行しているアプリケーションとアプリケーションが使用しているファイル・システム）はファイルがどの計算機に格納されているかを意識せず、ファイル621、621'等に容易にアクセスすることができる。

【0211】第2に、統合ファイル名結合子611が保持している結合情報データ構造の構造（図9）から明らかのように、ファイル621、621'等のうち複数の異なるファイルが、同じ統合ファイル名を持つことができる。すでに説明したように、アプリケーションが、同じ統合ファイル名を有する複数のファイルに対して当該統合ファイル名を指定してアクセスすると、統合ポリシーにより適当なファイルにアクセスすることができる。例えば、あるプログラムの複数バージョンのファイルに同じ統合ファイル名を与えることと、また最新かつアプリケーションにおける計算機で実行できるバージョンをアクセスする統合ポリシーを設定することにより、当該統合ファイル名で最新かつアプリケーションにおける計算機で実行できるプログラムを自動的に選んで実行することができる。また複製の場合、アクセスの結果は当該統合ファイル名を持つすべての複製ファイルに反映される。従って、ユーザはあるファイルの複製や複数バージョンを、ファイル名を変えずに設けることができる。しかも、当該複製のどれか1つに変更を加えた場合でも、ユーザはその他の当該複製に当該変更を自分で反映する必要がない。当該複数バージョンのある場合、ユーザは複数バージョンの存在を意識する必要がなく適当なバージョンをアクセスすることができる。

【0212】図18の分散計算機環境では、統合ファイル名結合子は大型計算機603にただ1つ設けられているが、これは統合ファイル名結合子を大型計算機603上で実行することが効率的であることによる。その理由は、大型計算機は一般に他のデスクトップ機と比較して高性能でかつ連続運転され、また大型計算機には大容量の磁気ディスク等が設けられ、多数のファイルが格納されているためである。しかし、既に述べたように、本実施形態の統合ファイル名システムでは複数の統合ファイル名結合子が1つの統合ファイル名ツリーを構成することができるため、場合によってはデスクトップ機601、601'等および602、602'等の上のどれかでも統合ファイル名結合子を実行できる。異なる計算機

上で複数の統合ファイル名結合子を実行することにより、統合ファイル名結合子内の処理を負荷分散して高速化したり、計算機のどれかに障害が発生して、その計算機上の統合ファイル名結合子が停止した場合でも、アプリケーションが統合ファイル名ツリーを用いてファイル・アクセスを継続できるようにしたりすることができる。

【0213】図18の分散計算機環境は、デスクトップ機と大型計算機がLANで接続された構成であるが、当該デスクトップ機や大型計算機がすべて1台の並列計算機のノードであり、LANの代わりに内部ネットワークで接続されている場合、あるいは一部が並列計算機のノードであり、残りは並列計算機外のデスクトップ機（または別の並列計算機の各ノード）であり、LANおよび内部ネットワークで接続されている場合でも、まったく同じように統合ファイル名システムを適用し、前記した計算機名（またはノード名）を指定する必要のないファイル・アクセスや自動的な複製管理とバージョン管理などの利点が得られる。

【0214】（応用例2）図19に、本応用例における分散計算機環境を示す。広域ネットワーク（WAN）700に、多数のサーバ計算機701、701'等が接続されている。当該サーバ計算機は、それぞれ大型計算機や、サーバ機や、あるいはデスクトップ機などの計算機である。また、WAN700にはデスクトップ機702やゲートウェイ705が接続されている。ゲートウェイ705は、後述するプロキシサーバ714を実行することのできる計算機（ルータ等も含む）である。ゲートウェイ705は、さらにローカル・エリア・ネットワーク（LAN）600に接続されている。LAN600には1台以上のデスクトップ機703、703'等が接続されている。LAN600にも1台以上のサーバ計算機704、704'等が接続されている場合がある。また、図示していないが、デスクトップ機702やゲートウェイ705と同様のデスクトップ機やゲートウェイが、WAN700に接続されていてもよい。

【0215】上記構成を有する図19の分散計算機環境では、下記のソフトウェアが実行されている。サーバ計算機701では、ファイル・サーバ713が実行されている。ファイル・サーバ713は、WAN700経由で要求を受信し、要求に応じて1つ以上のファイル721、721'等の内容を送信する。これらは、他のサーバ計算機701'等、および、LAN600経由で要求を受信することを除けば、サーバ計算機704、704'等（図19ではファイル・サーバの記載を省略している）でも同様である。また、デスクトップ機703および702では、ユーザによってアプリケーション712および712'が実行されている。アプリケーション712は、サーバ計算機701のアドレスと、サーバ計算機701が保持するファイル721のファイル名（サ

サーバ計算機701内でファイル721を識別できるファイル名)を指定して要求を送信することにより、ファイル721の内容を受信する。以下、デスクトップ機703'等においても、デスクトップ機703と同様である。

【0216】さらに、ゲートウェイ705では、プロキシサーバ714が実行されている。プロキシサーバ714は、アプリケーション712とファイル・サーバ713とを中継する役割を果たす。より具体的には、プロキシサーバ714は、アプリケーション712がファイル・サーバ713のファイル721を要求した時に、ファイル・サーバ713から転送されてきたファイルの内容を受信し(743)、自分の中のキャッシュ722に格納し、受信したファイルの内容を要求を送信したアプリケーション712に送信する(744)。これにより、アプリケーション712や、デスクトップ機703、703'等の上で実行されている別のアプリケーションが、同じファイル721を要求しようとしたときは、プロキシサーバ714のキャッシュ722の内容が転送されるため、WAN700を経由したファイル転送が発生せず、前記アプリケーションは要求したファイルを高速に受信することができる。なお、図示していないが、キャッシュ722には複数のファイルの内容を格納しているもよい。

【0217】上記構成の分散計算機環境における問題は、次の2点である。第1に、アプリケーション712は、ファイル721を指定するために、サーバ計算機701のアドレスをまず指定しなくてはならない。そのため、ファイル721の内容が必要な場合、ファイル721を格納しているサーバ計算機701のアドレスも知っていないだけではならず、例えばサーバ計算機の構成変更などによりファイル721が別のサーバ計算機に移動すると、アプリケーション712はファイル721を受信することができなくなる。第2に、プロキシサーバ714のキャッシュ722の内容は、多くの場合、ファイル721の内容と常に一致していることが要求される。図19の分散計算機環境でこの要求を満たすには、プロキシサーバ714は、アプリケーション712がファイル721を要求した時に、前記のようにキャッシュ722の内容をアプリケーション712に転送する前に、サーバ計算機701に対して、キャッシュ722の内容とファイル721の内容が一致しなくなったかどうかを毎回問い合わせ、もし不一致であれば、キャッシュ722の内容を破棄してサーバ計算機701からファイル721を転送しなくてはならない。この場合、アプリケーション712がファイル721を要求してからファイル721の内容を受信するまでの応答時間が長くなるという問題がある。

【0218】図19の分散計算機環境における上記の問題は、統合ファイル名システムを次のように適用するこ

とで解決できる。図19で、各サーバ計算機701および704上では、それぞれ統合ファイル名結合子711、711'が実行されている。また、図示していないが、その他のサーバ計算機701'等および704'等においても同様に統合ファイル名結合子が実行されていることができる。それぞれの統合ファイル名結合子は、すでに図7ないし図15を用いて説明した方式により領域名にもとづいて設定され、統合ファイル名ツリー2を構成している。上記の状況で、アプリケーション712、712'等およびプロキシサーバ714は、統合ファイル名を指定することにより、統合ファイル名ツリー2にもとづいて各サーバ計算機のファイルにアクセスすることができる。ここで、既に述べたように、ファイル721を指定する統合ファイル名に、ファイル721を格納しているサーバ計算機701のアドレスは含まれない。従って、アプリケーションがファイル721を必要とするとき、サーバ計算機701のアドレスを知っている必要はないだけでなく、ファイル721が別のサーバ計算機に移動した場合でも、アプリケーションは同じ統合ファイル名を指定することにより、ファイル721の内容を受け取ることができる。

【0219】また、プロキシサーバ714が前記のようにファイル721の内容を受信し、キャッシュ722に格納した後で、自分のアドレスを統合ファイル名結合子711に送信し、ファイル721の複製を保持していることを統合ファイル名結合子711に通知する(745)。これにより、すでに説明したように、統合ファイル名結合子711内の結合情報データ構造(図19には示していない)に、図10のファイル・サーバのアドレスとしてプロキシサーバ714のアドレスが格納され、統合方式に「複製」が格納された結合ユニットが新規に作成され、以後ファイル721が変更された場合、キャッシュ722の内容も更新され、常にファイル721の内容とキャッシュ722の内容が一致するように保たれる。従って、アプリケーション712がファイル721の内容を要求し、プロキシサーバ714内のキャッシュ722にファイル721の内容のコピーがある場合、プロキシサーバ714はキャッシュ722の内容がファイル・サーバ713にファイル721の内容と一致しているかどうかを調べることなくキャッシュ722の内容をアプリケーション712に転送することができ、アプリケーション712がファイル721を要求してからファイル721の内容を受信するまでの応答時間を短縮できる。

【0220】変形例として、アプリケーション712の実行されているデスクトップ機703自体が、過去に受け取ったファイルの内容をキャッシュとして保持している場合もありうる。この場合、デスクトップ機703上でファイル・サーバを実行し、アプリケーション712がファイルを受け取ってキャッシュに格納したときに、

前記ステップ745と同様に複製を有することを統合ファイル名結合子711に登録することにより、デスクトップ機703上で実行されるアプリケーションは、統合ファイル名を用いて常にファイル721の最新の内容を、自分と同じデスクトップ機内のキャッシュから高速に受け取ることができる。

【0221】また、前記の特別な場合として、統合ファイル名の付け方を定めることにより、各サーバ計算機の保持する情報を、分野ごとに一覧することができるようになる。前記の、ファイルを要求する際にアプリケーションがサーバ計算機のアドレスを指定しなくてはならない方法では、一般に1つの分野でも（例えば同業の企業ごとに）複数のサーバ計算機に分かれているため、ある分野の情報をまとめて受信することができない。それに対し、統合ファイル名システムでは、統合ファイル名を用いて、サーバ計算機のアドレスを特定せずにファイルを要求することができる。そこで、統合ファイル名の付け方として、それぞれのファイルの内容が属する分野を統合ファイル名に反映（例えば“//計算機/ソフトウェア/ファイルシステム/分散ファイルシステム/統合ファイル名システム”などと）させることにより、ある分野の情報を複数のサーバ計算機が保持している場合でも、アプリケーションは（例えば統合ファイル名“//計算機/ソフトウェア/ファイルシステム”などと指定することにより）所望の分野の情報を一覧することができる。

【0222】以上に示したように、本実施の形態によれば、従来技術のできない上下関係のないディレクトリマウントを含む多種のディレクトリマウントまたはファイルマウントを実現できる。また、異なるローカルファイル名ツリーに分散して配置されたディレクトリまたはファイルを一つのディレクトリまたはファイルにまとめることができる。しかも、統合ポリシーにより、特に同名によって重なった複数のファイルに対して、マウント時のマウント仕方およびマウント後のアクセス手順と一致制御方法とアクセス制御をユーザが設定可能となる。また、従来より便利な複製管理やバージョン管理などを行うことができる。

【0223】なお、以上に示したファイル統合と統合ポリシーを用いれば、ユーザまたはアプリケーションは、ファイル名を変えることなくファイルの複製や複数バージョンなどを設けることもできる。しかも、これらのファイルに対して一致制御方法とアクセス方法の選択は設定された統合ポリシーにより自動的に行われる。

【0224】

【発明の効果】本発明によるマウント方法によれば、ディレクトリのマウント時にディレクトリ毎にマウントを許可するファイルに関連する、ユーザが指定したマウント条件を設定できるので、いろいろの態様でマウントを行うことができる。

【0225】さらに、本発明によるファイルアクセス方法によれば、マウント先のディレクトリに対応して、ユーザが指定した読み出し操作および書き込み操作を設定できるので、ファイルアクセスをユーザ指定の態様により実行することができる。

【0226】さらに、本発明によるアクセス権限判別方法では、マウント先のディレクトリに対応して、ユーザ指定のアクセス権限情報を指定することができる。さらに、同じマウント先に対して、マウント元に対応してユーザが指定したアクセス権限情報も設定できる。

【図面の簡単な説明】

【図1】本発明の一実施の形態としての分散ファイルシステム統合ファイル名システムの、ハードウェア構成およびソフトウェア構成を示す図である。

【図2】従来のユニオン・マウントを示す図である。

【図3】本発明のマウント方法を示す図である。

【図4】統合ファイル名システムにおける、統合ファイル名ツリーの構築過程を示す図である。

【図5】統合ファイル名システムにおける、統合ファイル名ツリーの構築過程を示す図である。

【図6】統合ファイル名システムにおける、統合ファイル名ツリーの構築過程を示す図である。

【図7】統合ファイル名システムにおいて、統合ファイル名ツリーが複数の統合ファイル名結合子によって構築されているときの、ハードウェアおよびソフトウェア構成の一例を示す図である。

【図8】統合ファイル名システムにおける、アプリケーション、統合ファイル名結合子およびファイル・サーバの各プログラムの相互作用を示す図である。

【図9】統合ファイル名システムにおける、結合情報データ構造の内容を示す図である。

【図10】統合ファイル名システムにおける、結合ユニットの内容を示す図である。

【図11】統合ファイル名システムにおける、結合情報の内容を示す図である。

【図12】統合ファイル名システムにおける、結合情報データ構造の内容の一例を示す図である。

【図13】統合ファイル名システムにおける、結合情報データ構造の内容の一例を示す図である。

【図14】統合ファイル名システムにおける、結合情報データ構造の内容の一例を示す図である。

【図15】統合ファイル名システムにおける、結合情報データ構造の内容の一例を示す図である。

【図16】アクセス処理手順を示すフローチャートである。

【図17】図16におけるステップ535の動作の詳細を示す図である。

【図18】統合ファイル名システムの応用例1における、ソフトウェアおよびハードウェア構成を示す図である。

【図19】統合ファイル名システムの応用例2における、ソフトウェアおよびハードウェア構成を示す図である。

【図20】マウント処理手順を示すフローチャートである。

【図21】図20でのマウント可否判断処理のフローチャートである。

【符号の説明】

111、211-214、611、711 統合ファイ

ル名結合子

2、60 統合ファイル名ツリー

221-224 統合ファイル名ツリーの領域

320 統合ファイル名結合子リスト

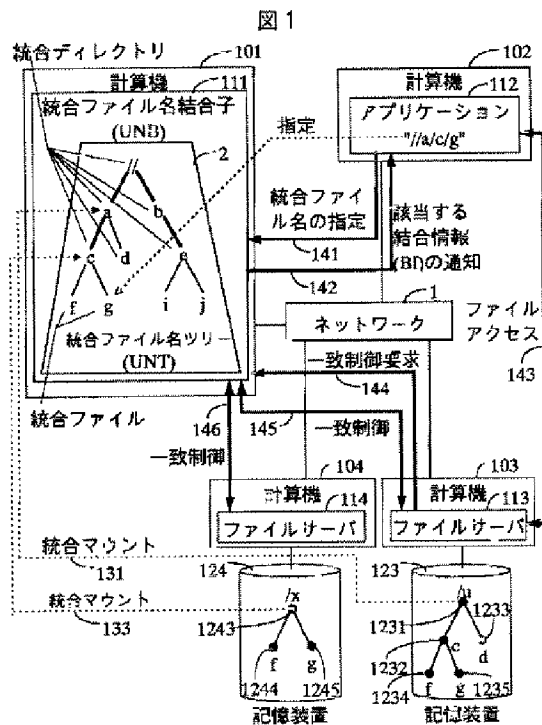
4、231-234 結合情報データ構造

411 結合情報

412 結合ユニット

421 結合子位置

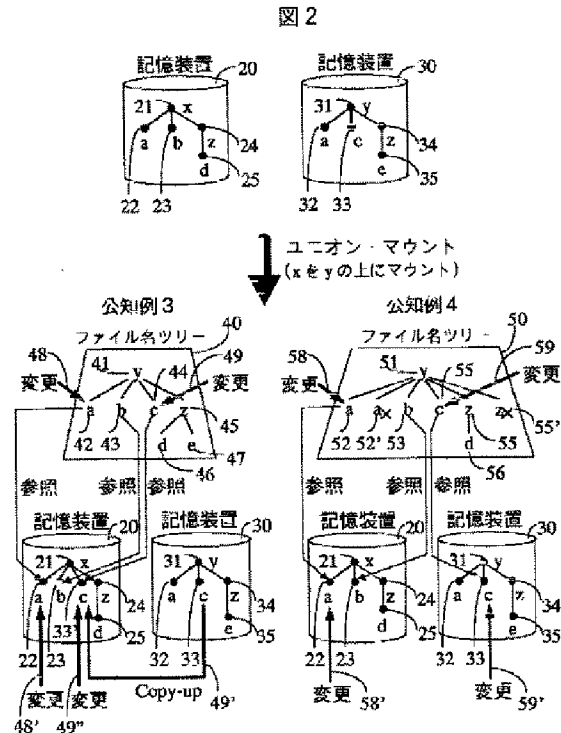
【図1】



○: ディレクトリ
またはファイルの実体

(磁気ディスク、光ディスク、
半導体メモリ、...)

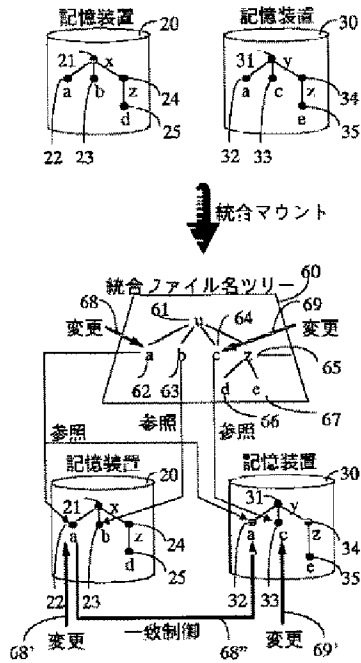
【図2】



○: ディレクトリまたはファイルの実体

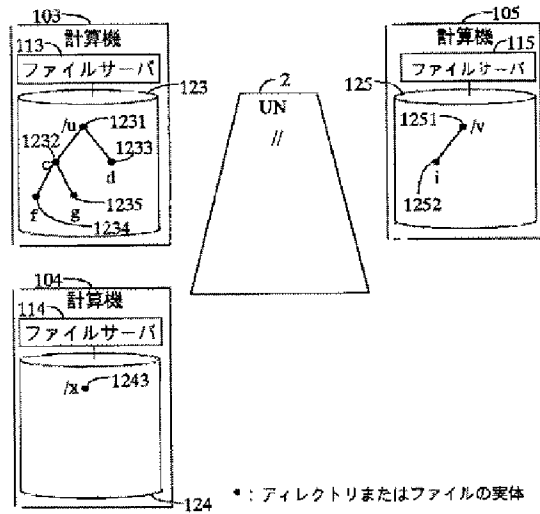
【図3】

图 3



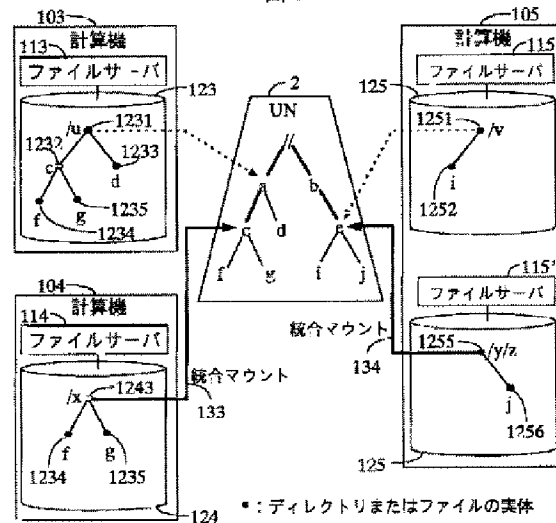
- : ディレクトリまたはファイルの実体

【図4】



- : ディレクトリまたはファイルの実体

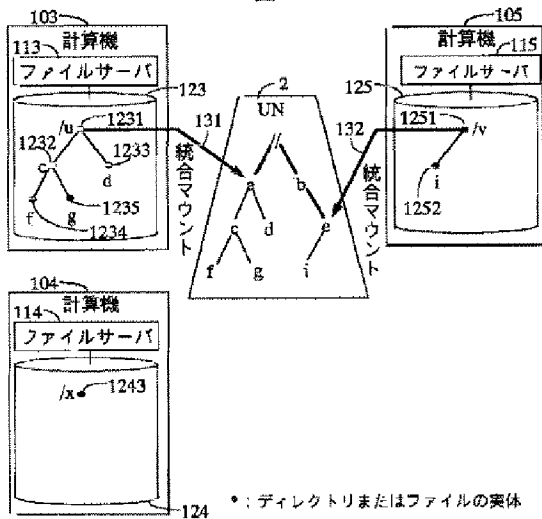
【圖6】

 6

- ・:ディレクトリまたはファイルの実体

【图5】

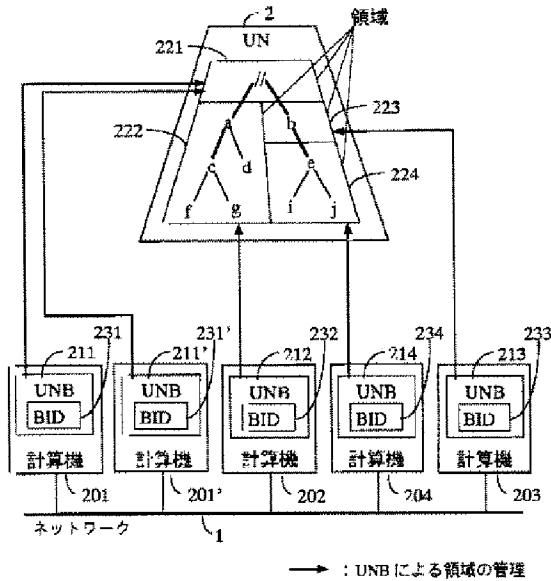
图 5



・:ディレクトリまたはファイルの実体

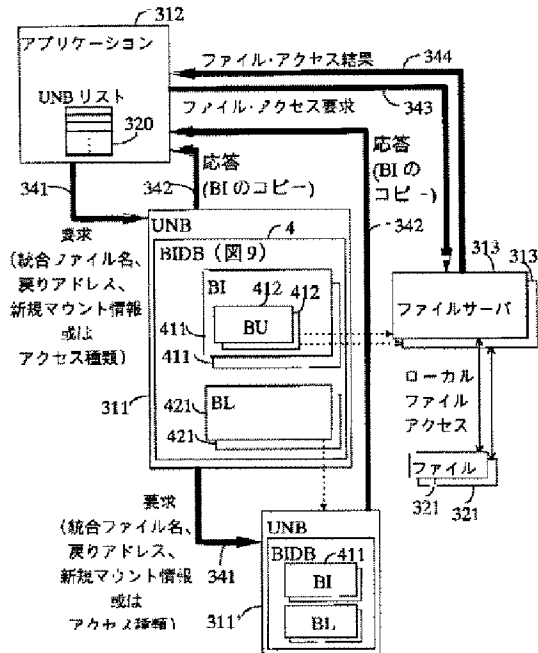
【図7】

図7



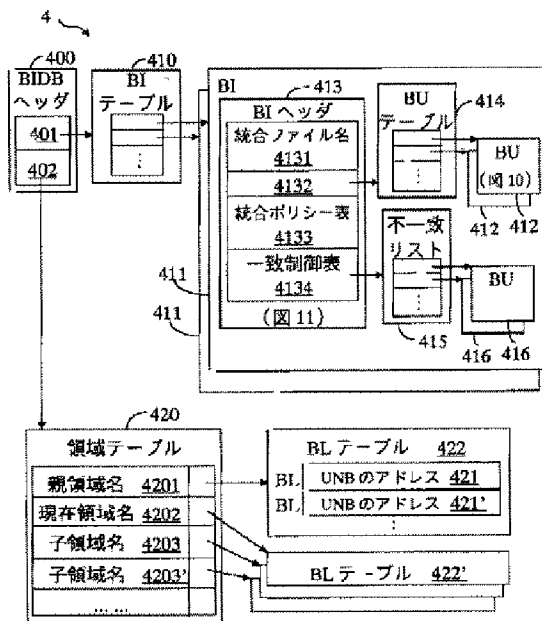
【図8】

図8



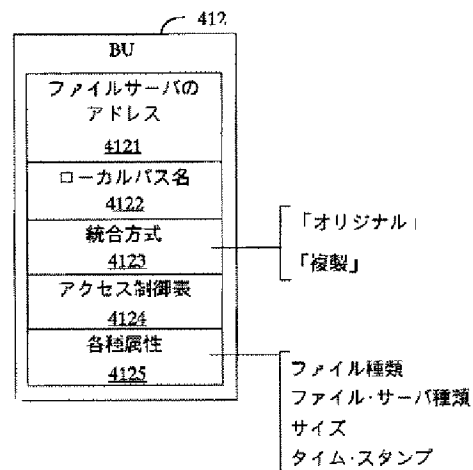
【図9】

図9



【図10】

図10



【図11】

【図12】

図11

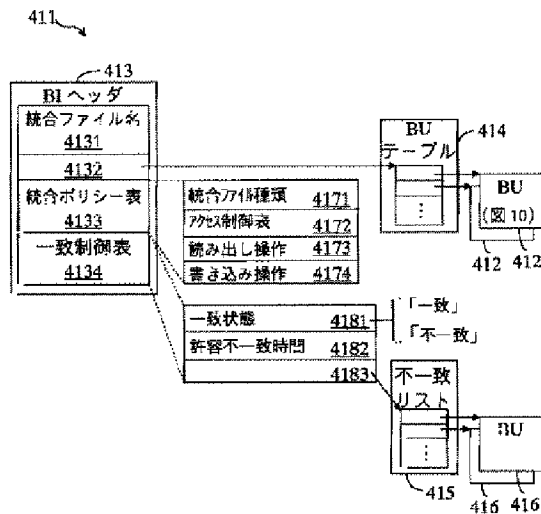
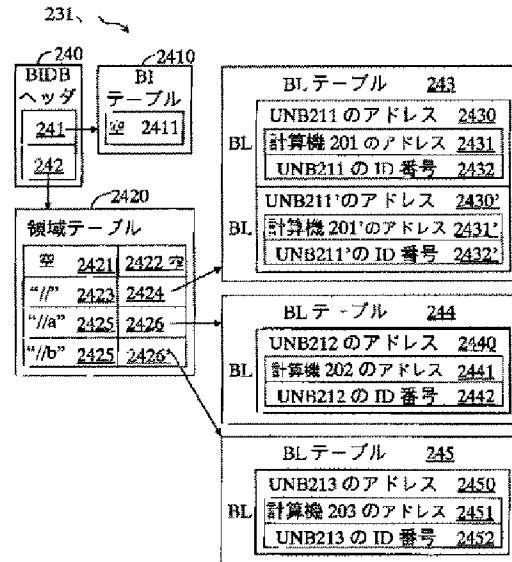


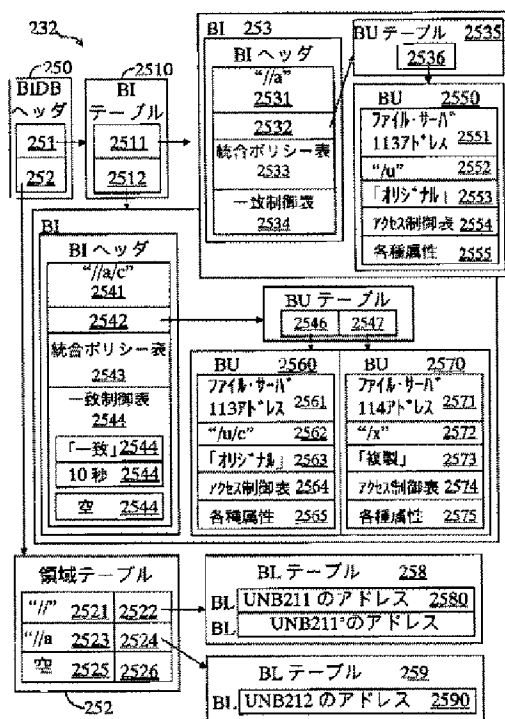
図12



【図13】

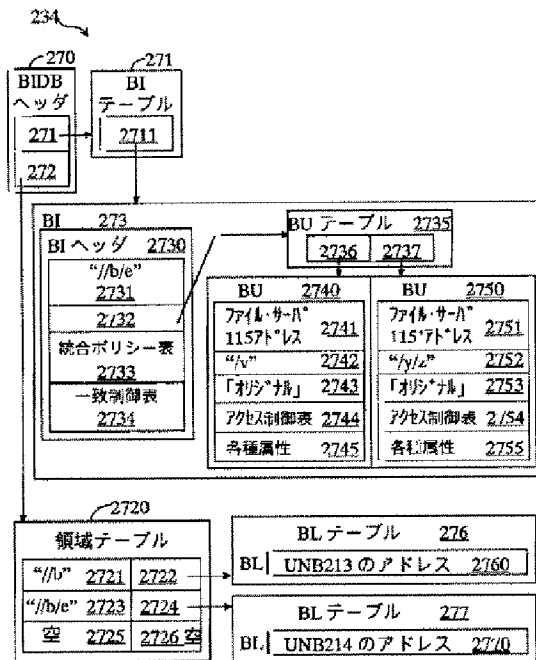
【図14】

図13



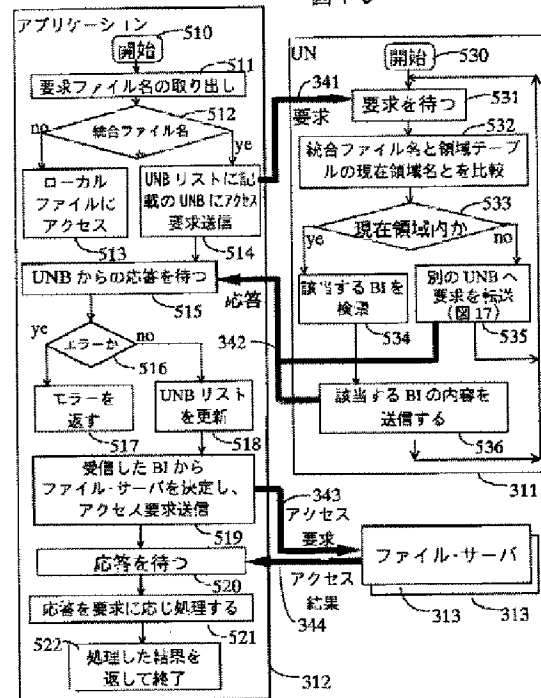
【図15】

図15



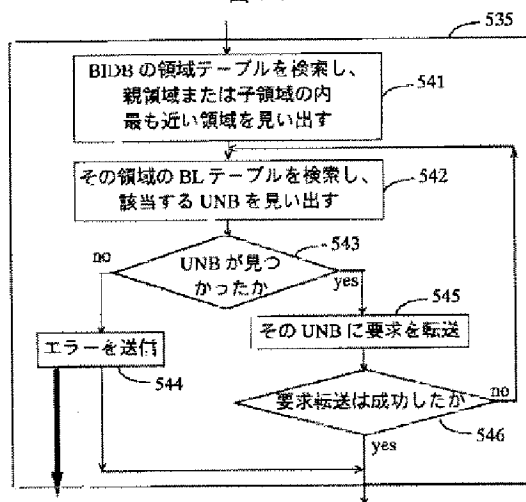
【図16】

図16



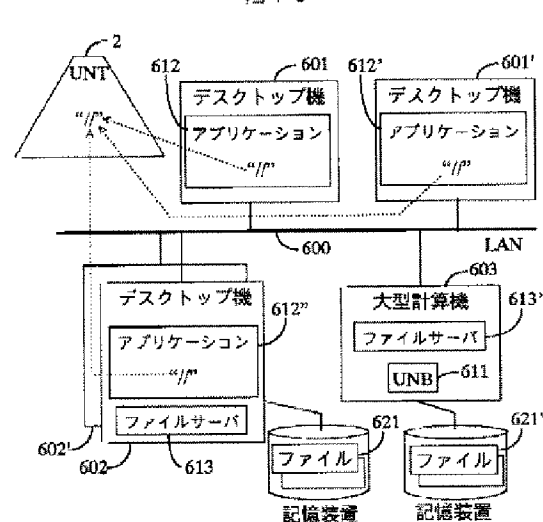
【図17】

図17

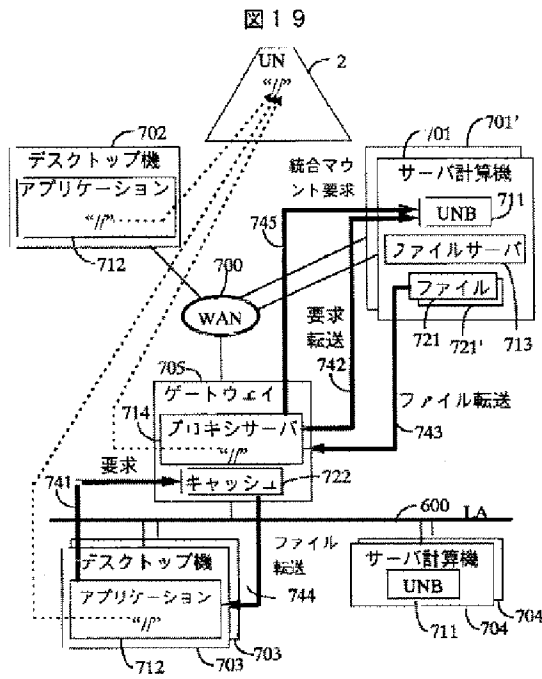


【図18】

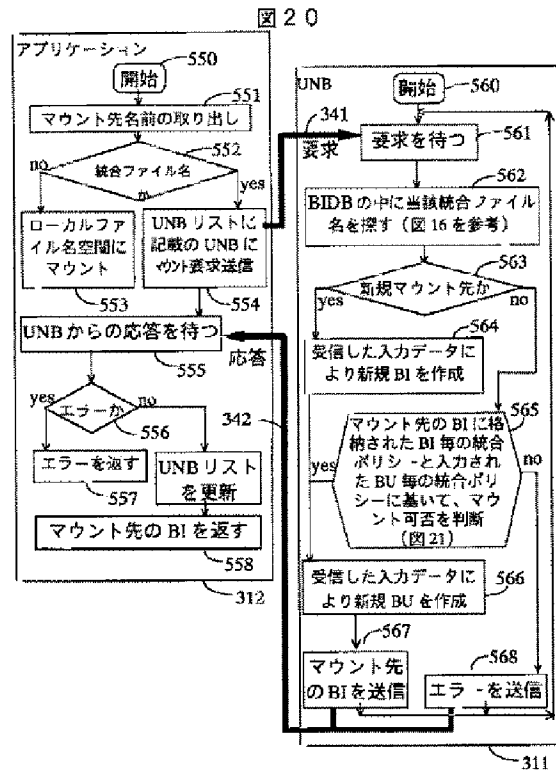
図18



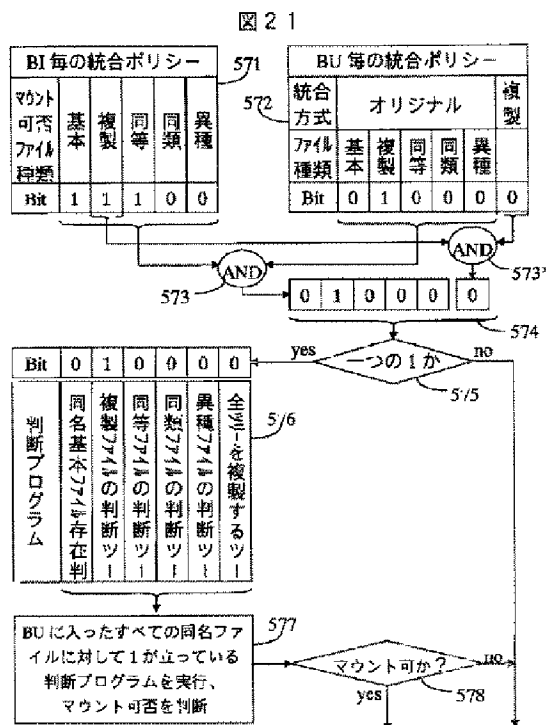
【図19】



【図20】



【図21】



フロントページの続き

(72)発明者 野田 文雄
東京都国分寺市東恋ヶ窪一丁目280番地
株式会社日立製作所中央研究所内